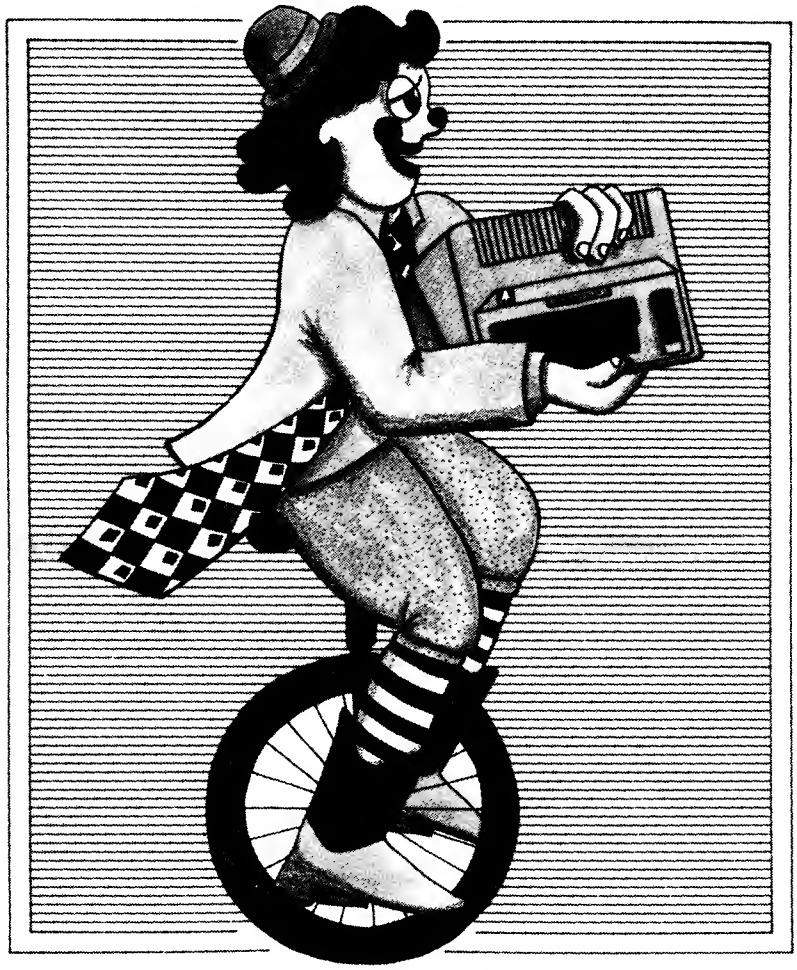
Sharon Boren • Larry Hovey • Kathleen Hovey

An ATARI® In The Classroom

Activity Workbook

dilithium Press

An ATARI® in



the Classroom

An ATARI® in the Classroom Activity Workbook

Sharon Boren

Larry Hovey

Kathleen Hovey

dilithium Press
Beaverton, Oregon

An ATARI in the Classroom Activity Workbook is part of a three-book set. As a workbook, it is not eligible to be catalogued by the Library of Congress. The following data applies to *An ATARI for Kids,* the principle text in the set.

Library of Congress Cataloging in Publication Data

Boren, Sharon, 1956–
An Atari for kids.

Includes index.
Summary: Teaches beginning programmers how to program the Atari or other microcomputers in the BASIC computer language.
1. Atari computer — Programming — Juvenile literature. 2. BASIC (Computer program language) — Juvenile literature.
(1. Atari computer — Programming. 2. Microcomputers — Programming. 3. BASIC (Computer program language) 4. Programming (Computers) 5. Computers) I. Hovey, Larry. II. Hovey, Kathleen. III. Title.
QA76.8.A82B67 1984 001.64'2 83-25269

ISBN 0-88056-123-8 (pbk.)

10 9 8 7 6 5 4 3 2 1

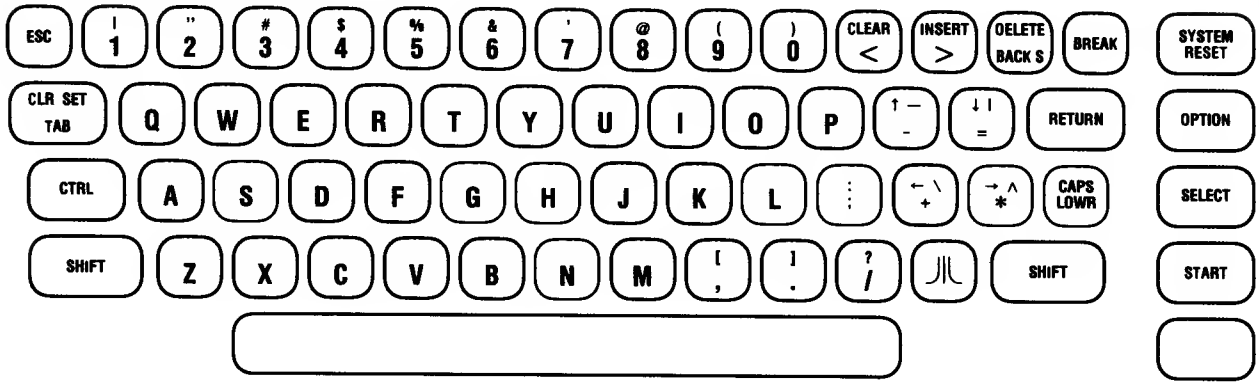
Library of Congress Cataloging in Publication Data

Printed in the United States of America

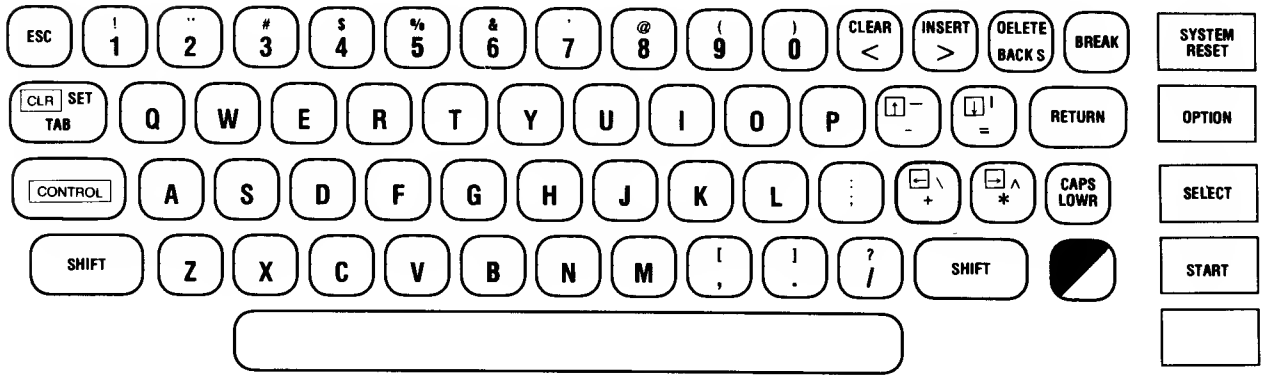
dilithium Press
8285 S.W. Nimbus
Suite 151
Beaverton, Oregon 97005-6401

TABLE OF CONTENTS

KEYBOARD ILLUSTRATIONS



ATARI 800



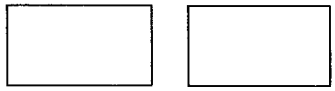
ATARI XL

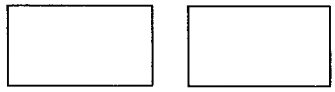
EXPLORING ATARI'S KEYBOARD #1

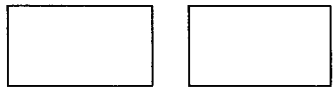
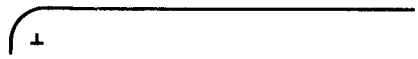
Take a few minutes to explore ATARI's keyboard. Press single keys, and keys while SHIFT CTRL are being held down, and see what happens. (The keys on the far right row are for commercial programs. Do not press them.)

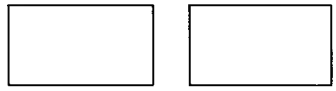
Finish drawing the key or keys that must be pressed to get ATARI to type what is shown on the screen at the right. Check your answers by using ATARI.

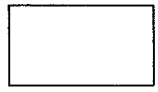
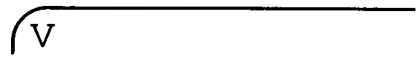
1. 

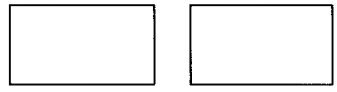

2. 

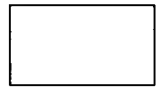

3. 

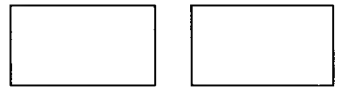

4. 


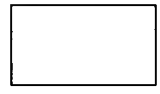
5. 


6. 


7. 


8. 


9. 


10. 


EXPLORING ATARI'S KEYBOARD #2

1. Turn ATARI on.

2. Press [RETURN].
   What did the cursor do? _____
   _____

3. Press [SPACE] (It's the long, unlabeled bar at the bottom.)
   What did the cursor do this time? _____

4. Press [SHIFT] and [CLEAR <] together.
   What did the cursor do? _____

5. Type your name (first, middle, and last).

6. Hold [CTRL] and press [← \ +] five times.
   Which way did the cursor move? _____
   Did anything happen to the writing on the screen? _____
   _____
   Now you know how to move the cursor around the screen without erasing any of the writing.

7. Hold [CTRL] and press the various cursor control keys (those with arrows [↑ −] [↓ | =] [← \ +] [→ ^ *]).
   Notice how the cursor moves around the screen without changing the writing.

8. Use the [CTRL] and cursor control keys (with arrows) to move the cursor to the first letter of your middle name.

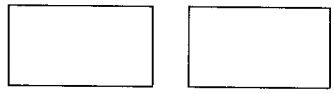
9. Press the space bar two times.
   What happened? _____
   _____

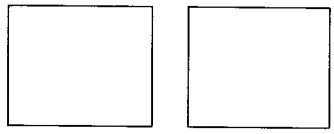
10. Press the [DELETE BACK S] key five times.
    What happened? _____
    _____

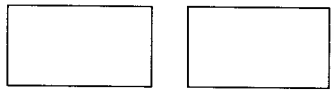
11. Hold [SHIFT] while pressing [CLEAR <] .
    What happened? _____
    _____

EXPLORING ATARI'S KEYBOARD #3

Finish drawing the key (or keys) that must be pressed to get ATARI to perform each special function.

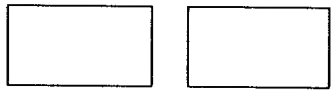
1.  Move the cursor up.

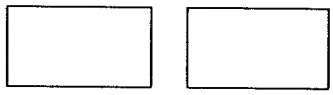
2.  Clear the screen and send the cursor home.

3.  Move the cursor right.

4.  Begin reverse field printing.

5.  End reverse field printing.

6.  Move the cursor left.

7.  Delete a letter.

OR 

8. ☐ ☐ Delete a line.

9. ☐ ☐ Insert a space.

10. ☐ ☐ Insert a line.

EXPLORING ATARI'S KEYBOARD #4

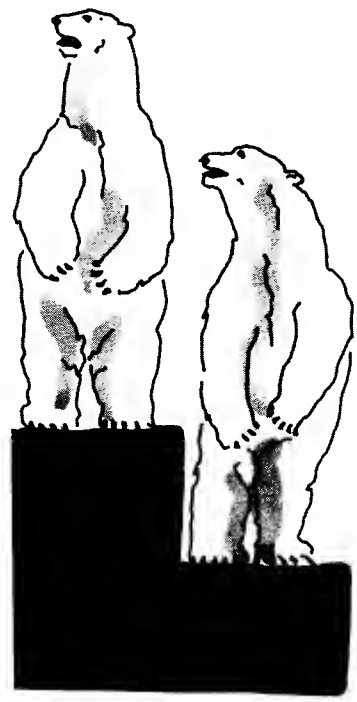
1. Turn ATARI on.
2. Type I LIKE YOU ATARI.
3. Press [DELETE BACK S] five times.
   What happened? _____
   _____
4. Hold [SHIFT] and press [INSERT >] five times.
   What happened? _____
   _____
5. Hold [SHIFT] and press [DELETE BACK S] five times.
   What happened? _____
   _____
6. Hold [SHIFT] and press [CLEAR <] .
   What happened? _____
   _____
7. Type I LOVE YOU ATARI.
8. Hold [CTRL] and press [↑ \ +] until the cursor is on the Y of YOU.
9. Hold [CTRL] and press [DELETE BACK S] four times.
   What happened? _____
   _____
10. Hold [CTRL] and press [INSERT >] four times.
    What happened? _____
    _____
11. Type YOU back in the new space.
12. If you have time, try typing some lines of your own, and use the various keys to do some screen editing.

EXPLORING ATARI'S KEYBOARD #5

Quick Review

Tell what happens when these keys are pressed:

1. [SHIFT] [CLEAR <] _____
2. [SHIFT] [INSERT >] _____
3. [SHIFT] [DELETE BACK S] _____
4. [CTRL] [INSERT >] _____
5. [CTRL] [DELETE BACK S] _____
6. [CTRL] [↑ −] _____
7. [CTRL] [↓ =] _____
8. [CTRL] [← \ +] _____
9. [CTRL] [→ ^ *] _____
10. [(SPACE)] _____
11. [DELETE BACK S] _____



EXPLORING ATARI'S KEYBOARD #6

Mine the Diamonds

created by Wendy Cheldelin

1. Turn ATARI on.

2. Clear the screen.

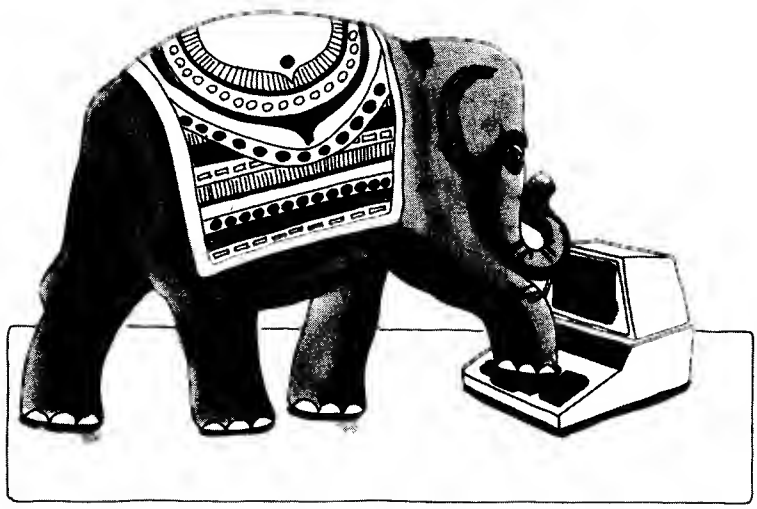
3. Press [CTRL] and hold it down. Press the [P] key 5 times.

4. Press [CTRL] and hold it down. Now press the [!] key once. There should be 5 "rocks" and 1 "diamond" on your screen.

5. On the same line, make 4 or 5 more rocks followed by 1 diamond until the line is filled and the cursor has moved to the line below.

6. Fill one more line with rocks and diamonds.

7. Now the challenge begins! Mine the diamonds by erasing all of the rocks. Be careful! Don't erase any diamonds.



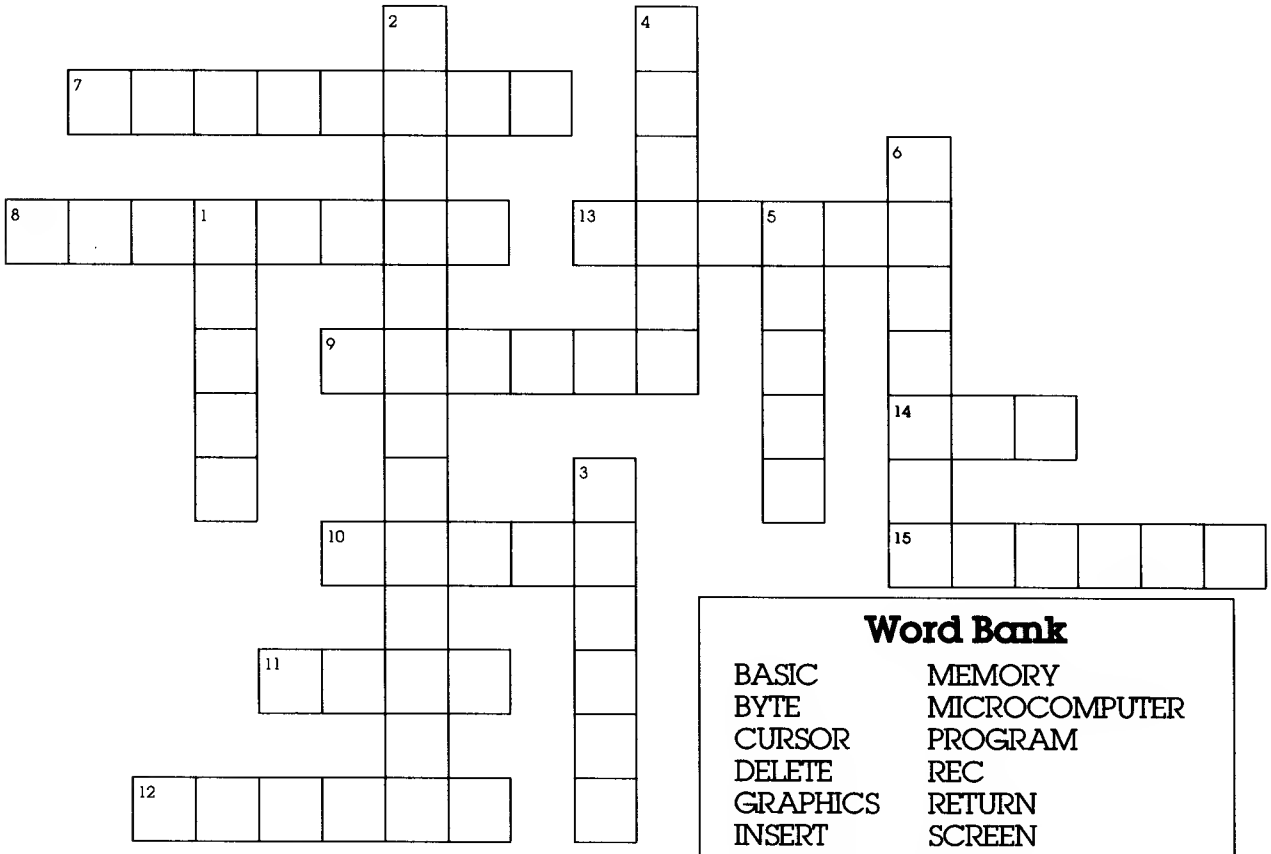
EXPLORING ATARI'S KEYBOARD #7

Design Your Own Game!

1. Design a game that requires some understanding of ATARI's keyboard. Write clear directions on how to play your game in the space below.

2. Ask a friend to play your game!

COMPONENT 1 FUN PAGE



Word Bank

BASIC	MEMORY
BYTE	MICROCOMPUTER
CURSOR	PROGRAM
DELETE	REC
GRAPHICS	RETURN
INSERT	SCREEN
KEYBOARD	SHIFT
	SPACE

down

1. The language that ATARI speaks.
2. What type of computer is ATARI?
3. What is another word for "erase"?
4. We must always press this key when we are done typing a line.
5. Hold this key down as you press another key and ATARI will print the symbol at the top or front of the key.
6. The set of directions a computer uses.

across

7. Symbols used to make pictures and borders.

8. Something that both a computer and a typewriter have.
9. The part of ATARI that shows what is typed.

10. Always press this key between words and numbers that you type.

11. The space it takes to store one letter in ATARI's memory.

12. What is another word for "add"?

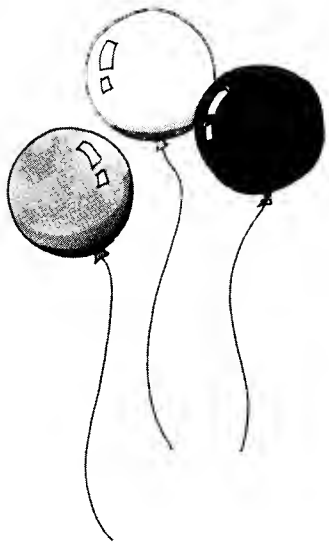
13. The white square that shows you where ATARI will type next on the screen.

14. Means "record".
15. The place where ATARI remembers programs.

Evaluate Yourself

1. In component 1, I did _____
   because _____
   _____.
2. Component 1 was _____.
3. Tell about the good parts of the component:
   _____
   _____.
4. Tell about the bad parts of the component:
   _____
   _____.

PROGRAMMING YOUR ATARI #1



Speak

1. Write a program that tells ATARI to print

   COMPUTER PROGRAMMING IS FUN!

2. Make sure each line of your program begins with a line number.

3. Check your program to make sure there are no mistakes.

4. RUN your program on the computer.

Use this format

```
10 PRINT "                "
20 END
```

Write your program here

Speaking Nonstop

1. Write a program that tells ATARI to print your name over and over again!
2. RUN your program on ATARI.

Use this format

```
10 PRINT "                "
   (Type your name inside the quotes.)
20 GOTO 10
```

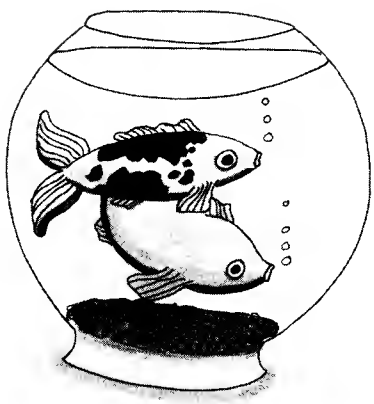
Write your program here

Top-Secret

1. Your mission is to write a program that tells ATARI to print a top-secret message in a secret code. Use the graphic symbols on the keys as your code. For example, if we wanted a word in our message to say SAW, the code would be + ⊢ ⊤ because these graphic symbols appear on the S, A, and W keys.
2. Give your program to a friend. Have your friend RUN it on ATARI and try to decode the secret message.

Your success as a secret agent depends on this program. Good luck! (This page will self-destruct in two days if your program is not finished.)

Write your program here

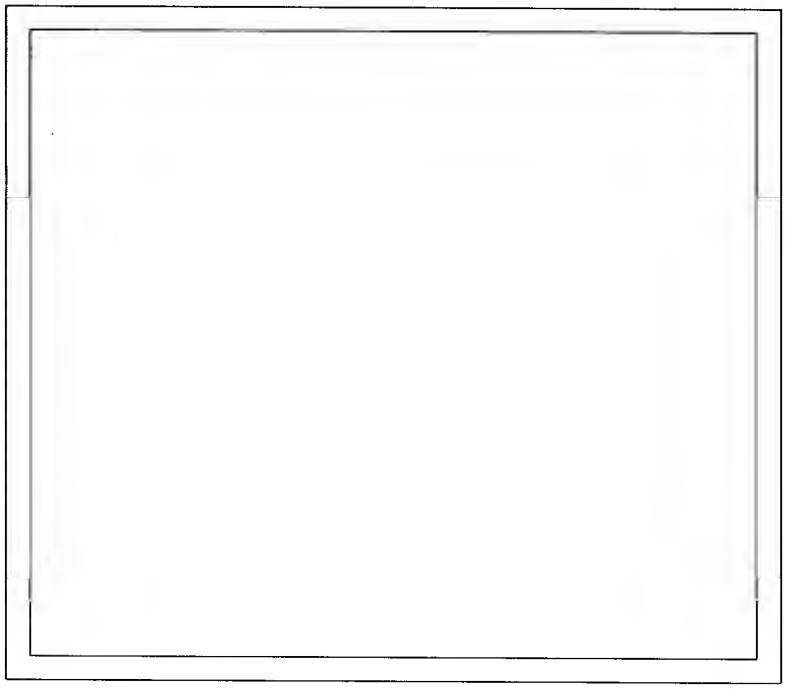


PROGRAMMING YOUR ATARI #4

Computer Art

1. Write a program that tells ATARI to print a design using graphic symbols.
2. Make it so that your design will be printed over and over on the screen.
3. RUN the program on ATARI.

Write your program here



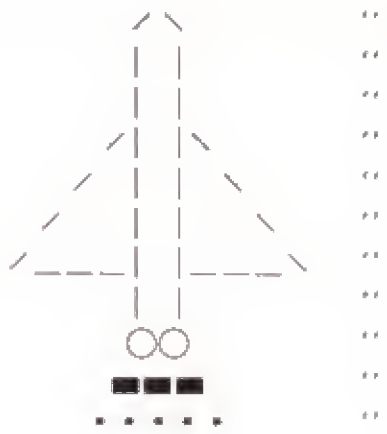
PROGRAMMING YOUR ATARI #5

Shuttle Launch

1. You can program ATARI to make moving pictures by using these five features:
   line numbers
   PRINT statements
   quotation marks
   GOTO statement
   graphic symbols
2. Use the format below to create a program that launches a rocket.
3. RUN the program on ATARI.

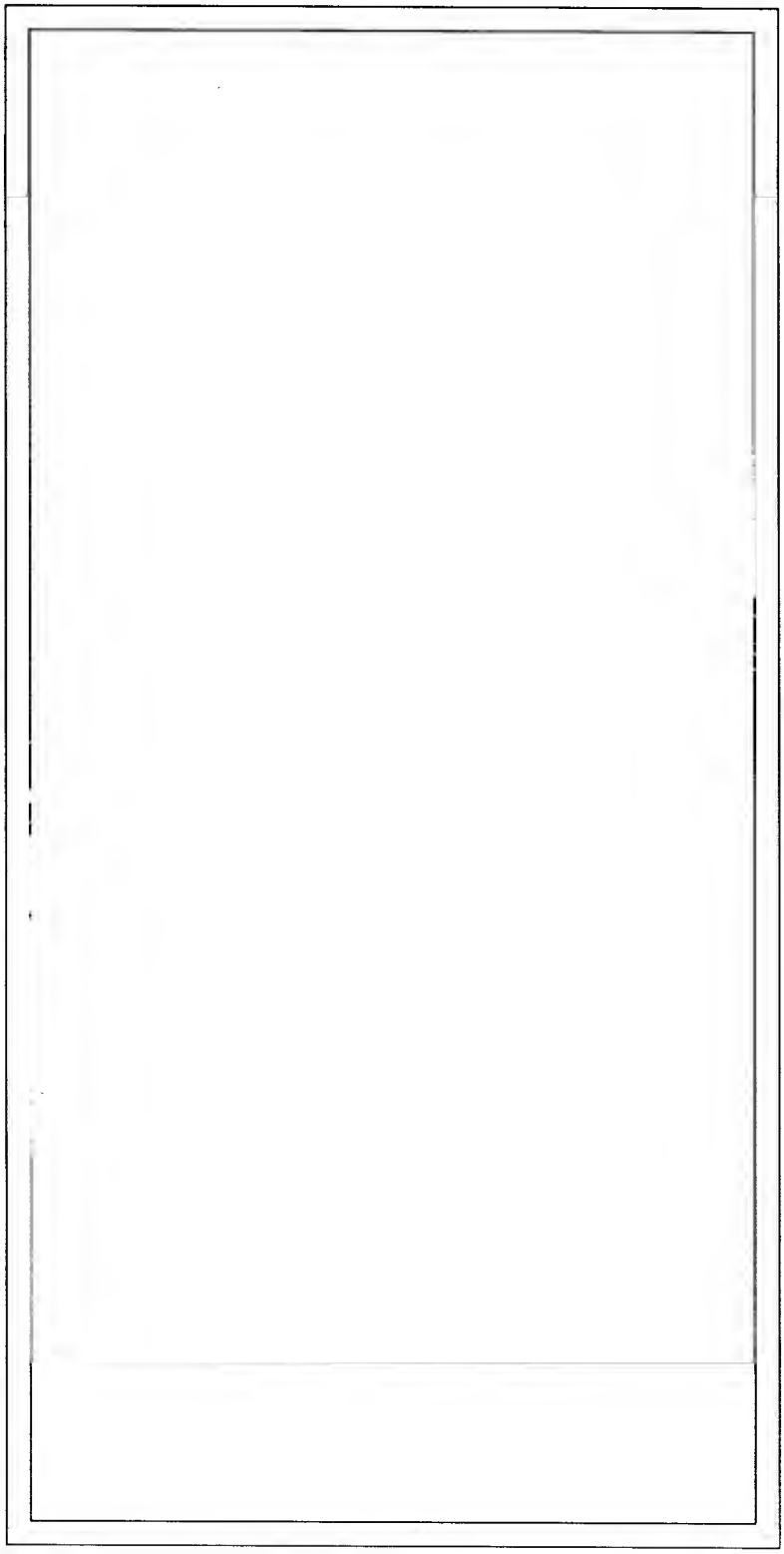
Use this format

```
10 PRINT "                "
20 PRINT "                "
30 PRINT "                "
40 PRINT "                "
50 PRINT "                "
60 PRINT "                "
70 PRINT "                "
80 PRINT "                "
90 PRINT "                "
100 PRINT "               "
110 PRINT "               "
120 PRINT "               "
130 PRINT "               "
140 GOTO 10
```



NOTE: It is important to leave lines 120 and 130 blank inside the quotation marks so the rockets are spaced out when they move on the screen.

Write your program here



You may have more than 140 lines in your program.

PROGRAMMER'S PASTIME #1

Write the symbol that ATARI uses for each arithmetic operation.

addition _____ multiplication _____
powers _____ subtraction _____
division _____ square root _____

How would you type each equation to get answers from ATARI?

1. $457+99\times6$ _____
2. $\sqrt{64}$ _____
3. $26\div2^2$ _____
4. $777\times555\div222$ _____
5. 8^3-16 _____
6. $\sqrt{22}\div88$ _____
7. $\sqrt{49}+765$ _____
8. $98+88\times66\div2^4$ _____

Show how you would type the equations above using only one PRINT (?) statement.

_____
_____
_____
_____
_____
_____

PROGRAMMER'S PASTIME #2

What order does ATARI follow in doing arithmetic for equations with many numbers,

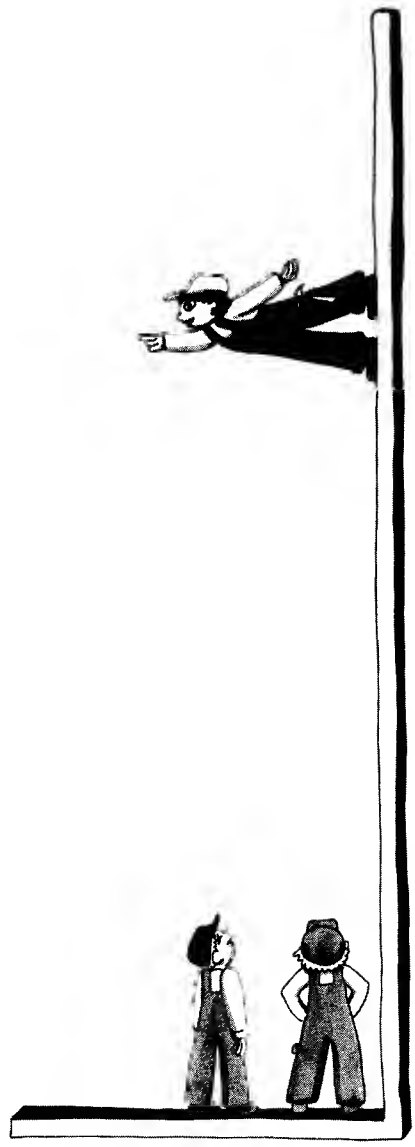
_____ are done first.
_____ are done second.
_____ and _____
are done third (left to right).
_____ and _____
are done last (left to right).

Use your mental powers and write the answers that ATARI would give for these equations. Remember to do the arithmetic in the same order that ATARI would!

1. 2*3+1 _____
2. 2+8*2*1 _____
3. 3*3+9+20 _____
4. 11+4*3 _____
5. 22+8+12*1 _____

Try some more.

1. 8/2−3 _____
2. 20/4+6−5 _____
3. 30−10/2 _____
4. 50−20/10+3 _____
5. 16/2+8/2 _____
6. 14/2+4/2−6 _____



PROGRAMMER'S PASTIME #3

Use your mental powers and write the answer ATARI would give for these equations.

1. 4*4+6/2 _____
2. 8/4+3*3 _____
3. 20−(4+5*2)+15 _____
4. 6+14/7*5 _____
5. (9/3−2)*(7*2+4) _____
6. (7+2−4+6*1)/1 _____

POWER!

Powers are also called EXPONENTS. Read the following examples and figure out how powers work.

1. 10∧1=10*1=10
2. 10∧2=10*10=100
3. 10∧3=10*10*10=1000

Try some more:

1. 2∧1=2*1=2
2. 2∧2=2*2=4
3. 2∧3=2*2*2=8
4. 2∧4=2*2*2*2=16

Identify the powers by filling in the blanks:

1. 3∧1=_____=_____
2. 3∧2=_____=_____
3. 3∧3=_____=_____
4. 3∧4=_____=_____

Now try these:

1. 4∧1=_____=_____
2. 4∧2=_____=_____
3. 4∧3=_____=_____
4. 4∧4=_____=_____

CHALLENGE

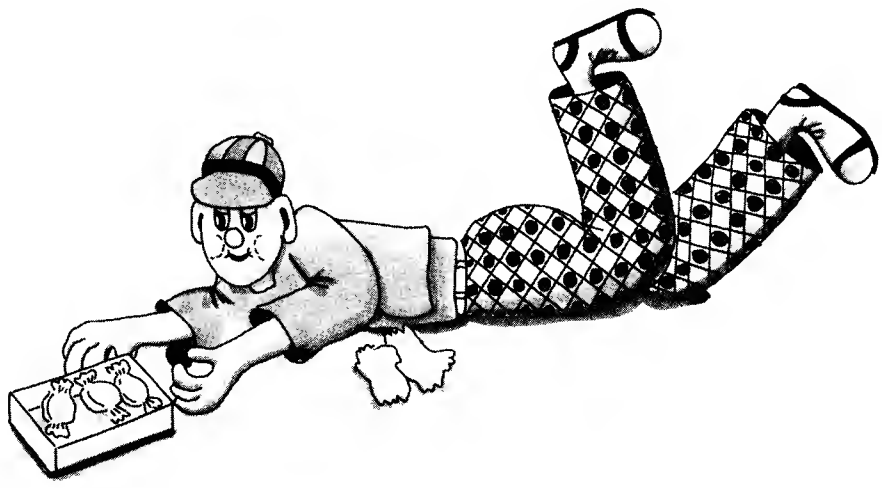
1. 2∧4∗5 _____
2. 5∗2∧4 _____
3. 7+3∧2 _____
4. 5∧2−13 _____
5. 100/10∧2 _____
6. 12∧2−5∗10 _____
7. 8∧2∗(4+5) _____
8. 200−(6+3∧3)+7 _____

(REMEMBER—If you check these on the computer, ATARI gives a slightly inaccurate answer when working with powers.)

In each of the following equations, put a 1 under the part the computer would do first, a 2 under the second part it would do, and so on.

Example: 82+72 / 2−(4+22)+9 ∧ 3
4 3 5 1 6 2

1. 3000−(13−6∗4)+8 ∧ 3+9
   _ _ _ _ _ _
2. 900 / 7 ∧ 2+(16−9 ∧ 4) ∧ 3
   _ _ _ _ _ _



PROGRAMMER'S PASTIME #4

Cowboy Clyde typed in the following equations, but ATARI wouldn't give him answers. Do you know why?

Find out what's wrong with the way his equations are typed. Write the correct way in the blanks.

1. ?62+4×20 _____
2. 23/4*6 _____
3. ?SQR 16 _____
4. 80/4+2×3*SQR 25 _____

CHALLENGE

If there's some arithmetic in a long equation that you want to be done *first*, put parentheses around it. (ATARI always does what's in parentheses first.) Let's say you want 4+3*2 to equal 14. If you type: 4+3*2 ATARI will give you 10 because multiplication is done before addition. So it becomes 4+6 which equals 10.

If you want 4+3*2 to equal 14, you must use parentheses like this: (4+3)*2.

first

7*2=14

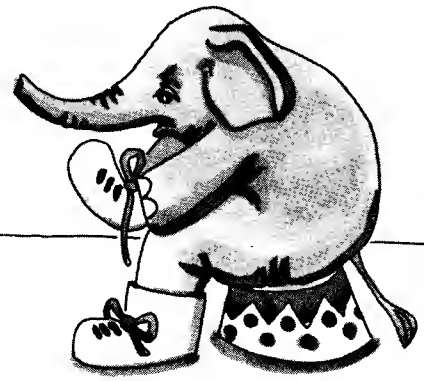
Rewrite each equation below and put parentheses around what ATARI should do first in order to make the equation TRUE. HINT: You might have to use *two* sets of parentheses for some equations.

Example: 7*3+2=35 — 7*(3+2)=35

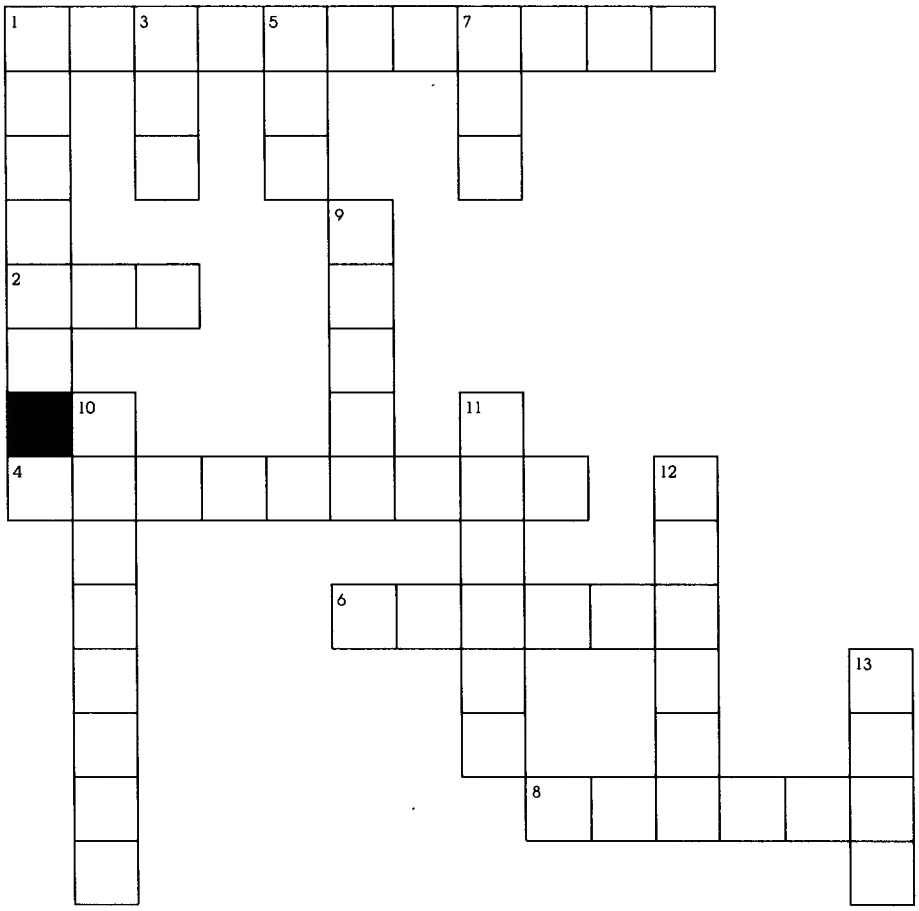
1. 9/2+1=3
2. 6*2+2=24
3. 3^3-1=9
4. 4+2-1*5=9
5. 12-3+6/3=9
6. 20-10^4+2=10002
7. 12/4+2=2
8. 9-5^2=16
9. 50+10/18+10+2=2

*10. 10+4-7^2*1+3-3=193

An * means it's an extra tough problem!



COMPONENT 2 FUN PAGE



Word Bank

COMMAS	PARENTHESES
DIRECT	POWERS
END	PRINT
LINE	QUESTION
NEW	QUOTATION
NUMBER	RETURN
	RUN (2)

Down

1. In a long equation, ATARI does _____ second.
3. Type the _____ command when you want to see ATARI do a trick or run a program.
5. Which command do you type when you want to clear ATARI's memory or do something new?
7. Which statement in a program tells ATARI that your program has ended?
9. Which program statement tells ATARI to write something?
10. A _____ mark is a shortcut used in place of a PRINT statement.
11. Many equations on one line must be separated by _____.
12. Using ATARI as a calculator is called _____ mode programming.
13. In a program,
    10
    20
    30 are called _____ numbers.

Across

1. In a long equation, ATARI does the operation in _____ first.
2. Which command do you type when ATARI is finished loading a game and ready to play?
4. Put _____ marks around what you want ATARI to say in a program.
6. Every step in a program must begin with a line _____.
8. Which key should you press when you are done typing a line and you want to go to the next line?

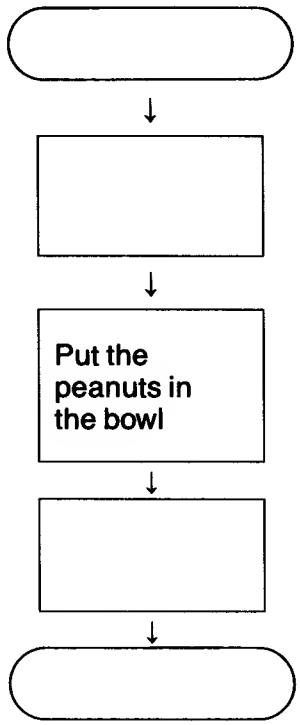
Evaluate Yourself

1. In component 2, I did _____
   because _____
   _____.
2. Component 2 was _____
   _____.
3. Tell about the best parts of the component:
   _____
   _____.
4. Tell about the parts of the component that were hard for you: _____
   _____
   _____.
5. Tell about the parts of the component that you like the least: _____
   _____
   _____.

PROGRAMMER'S PASTIME #5

For each flow chart, fill in the blank boxes with the step you think would fit. Make sure your steps are in the right order.

Algorithm/Flow Chart #1
How to feed your pet elephant.



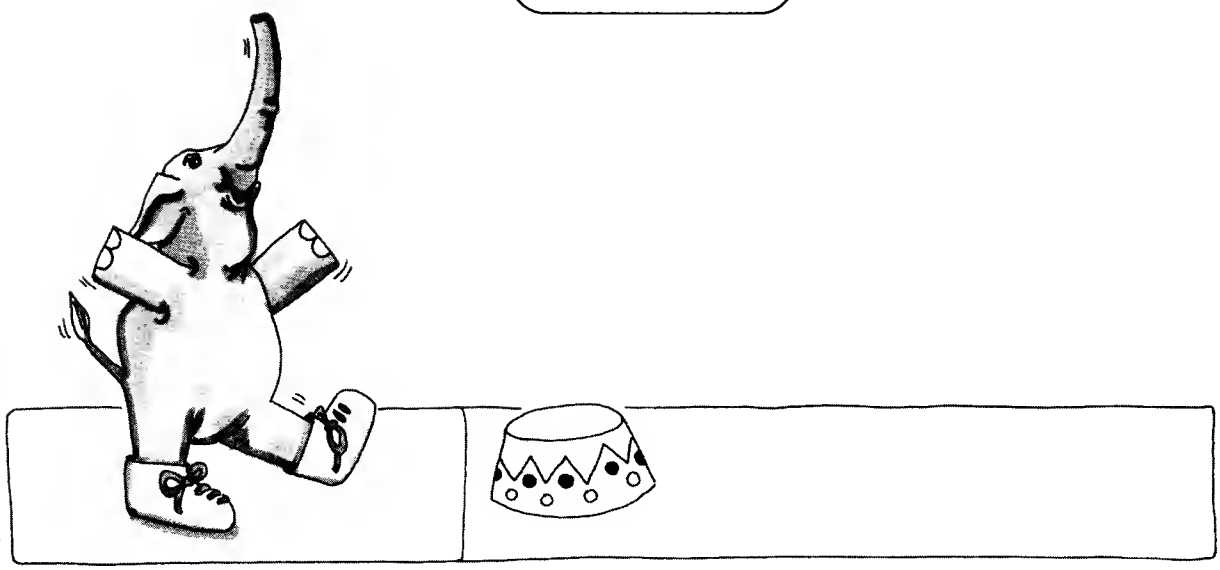
Missing Steps

STOP

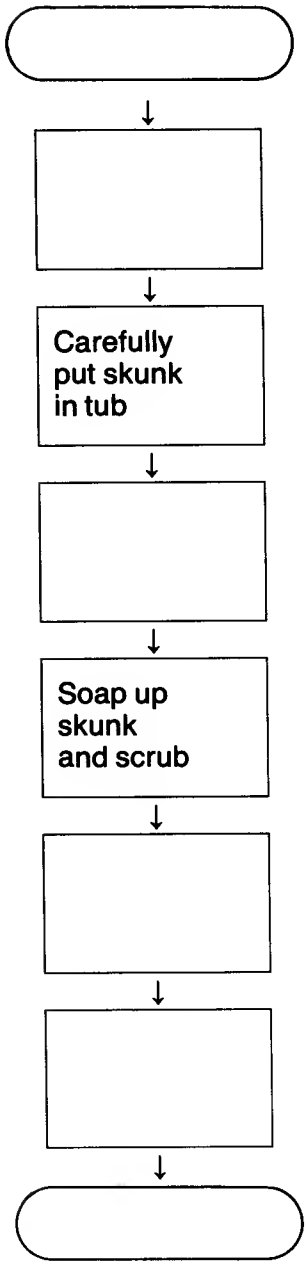
Call your pet elephant.

START

Get out the peanuts.



Algorithm/Flow Chart #2
How to wash your pet skunk.



Missing Steps

Get skunk wet

STOP

Rinse skunk

Fill tub with warm water

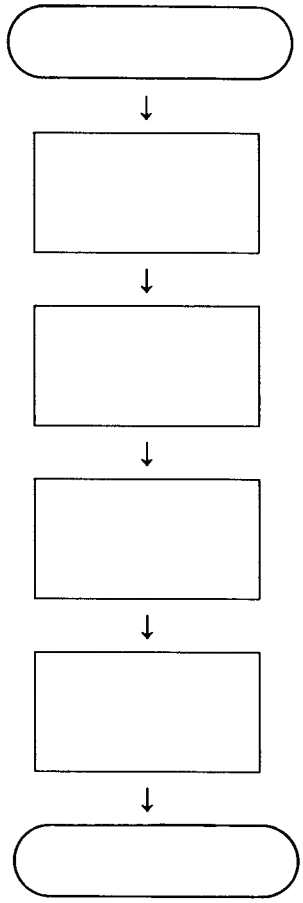
START

Dry skunk

PROGRAMMER'S PASTIME #6

For each flow chart, fill in the blank boxes with the step you think would fit. Make sure your steps are in the right order.

Algorithm/Flow Chart #1
How to walk your pet alligator.



Missing Steps

Attach leash to collar

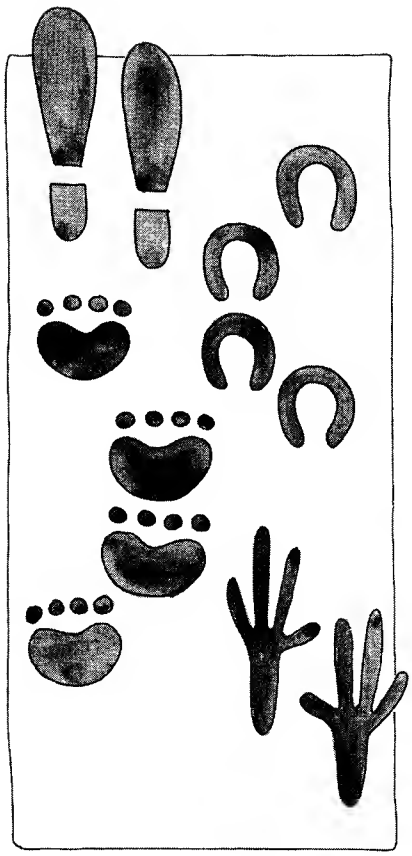
Go for a walk

START

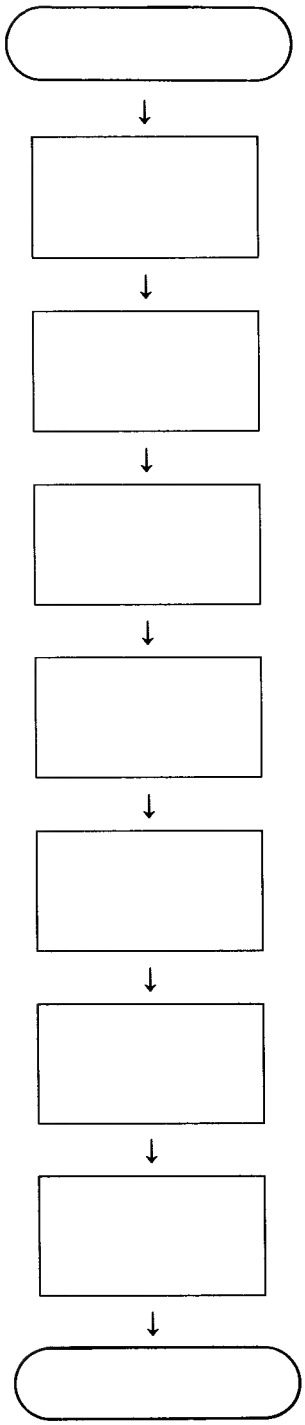
Muzzle the alligator

STOP

Put on his collar



Algorithm/Flow Chart #2
How to make an ice cream cone.



Missing Steps

Put away ice cream

Scoop out ice cream

START

Get out ice cream

STOP

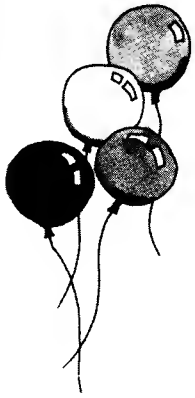
Get out a cone

Put scoop of ice cream into cone

Pack the ice cream firmly into cone

Eat ice cream cone

PROGRAMMER'S PASTIME #7

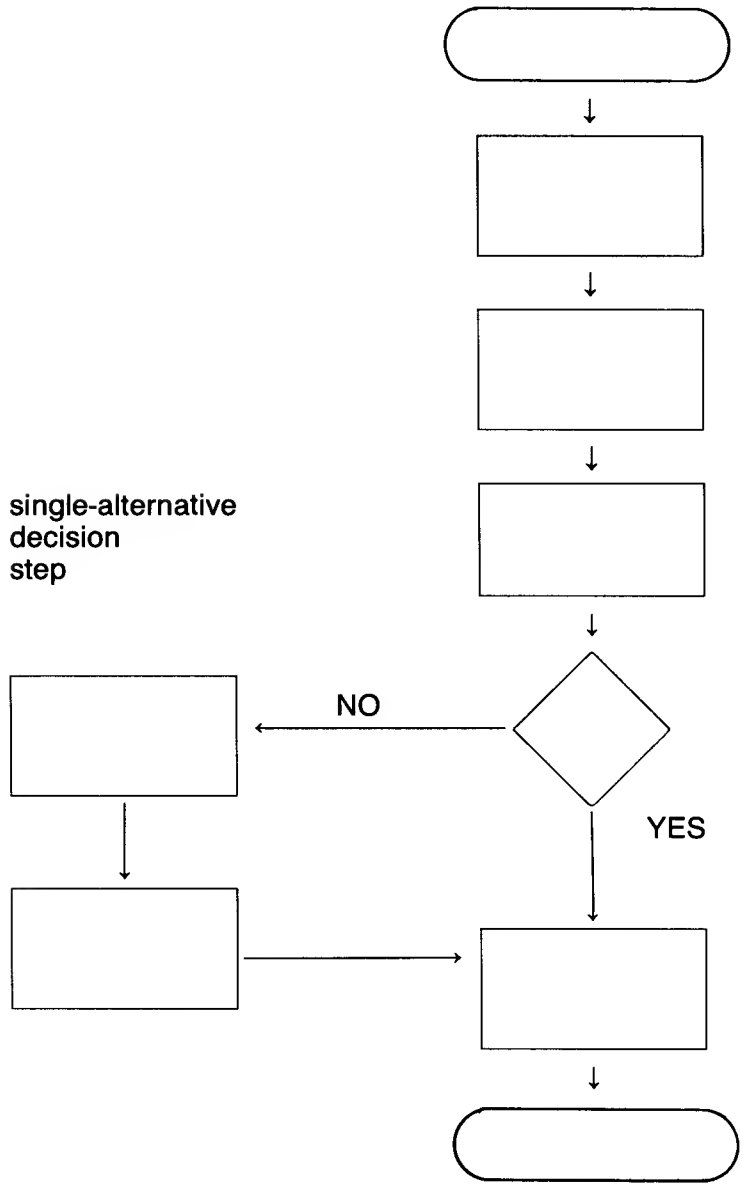


Design an algorithm for how to make a peanut butter and banana sandwich. Write your algorithm in flow chart form.

PROGRAMMER'S PASTIME #8

For each flow chart, fill in the blank boxes with the steps you think would fit. Make sure your steps are in the right order.

Algorithm/Flow Chart #1:
How to take a hot bath.



Missing Steps

Fill tub with water

Enjoy your bath

START

Turn off hot water

Turn off water

STOP

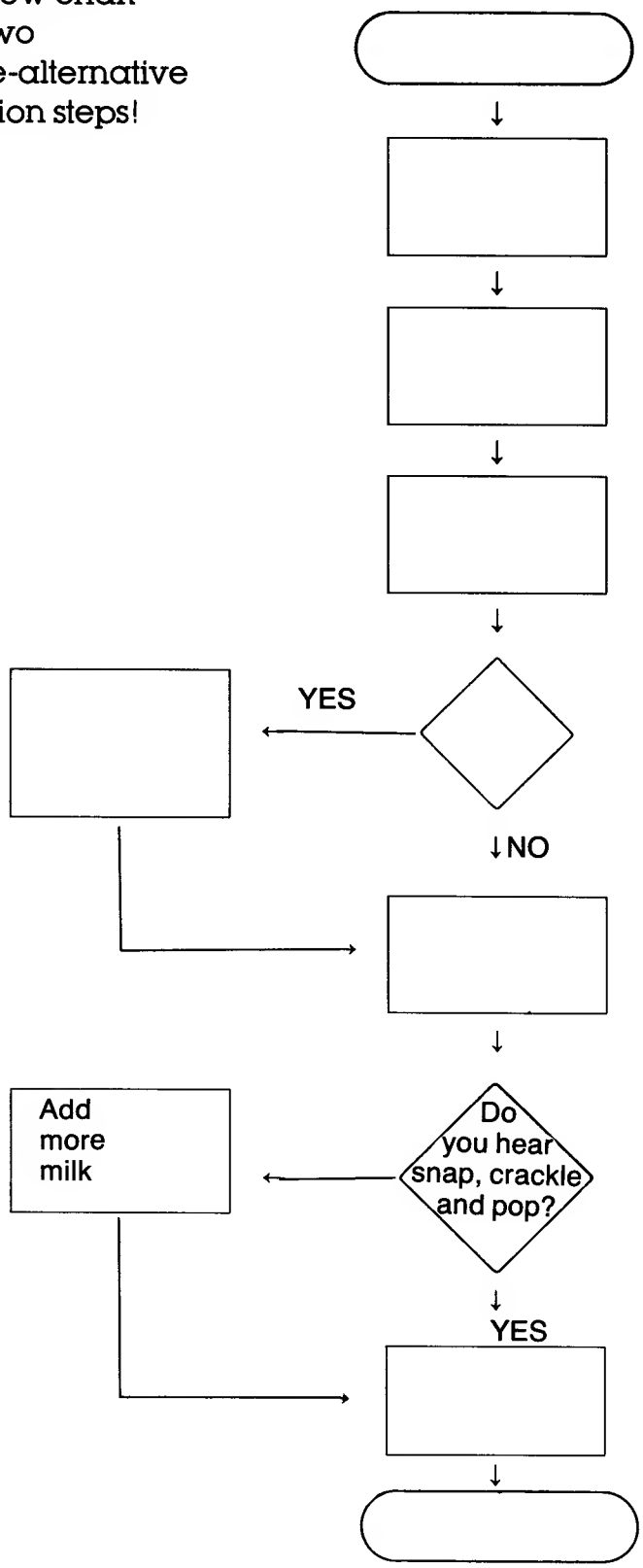
Is water hot enough?

Plug drain in tub

Turn on more hot water

Algorithm/Flow Chart #2
How to eat a bowl of Rice Krispies.

This flow chart has two single-alternative decision steps!



Missing Steps

Eat your cereal

Get out bowl & spoon

STOP

Add milk

Pour Rice Krispies into bowl

Is there too much cereal in your bowl?

Get out Rice Krispies and milk

Remove some cereal from bowl and put back into box

START

PROGRAMMER'S PASTIME #9

Here are the steps of an algorithm to make a chocolate milkshake. Put the steps in order and make a flow chart. Be sure to show the SINGLE-ALTERNATIVE DECISION STEP.

Steps

STOP

Add 1/8 cup chocolate sauce

Get out blender

Put 3 scoops of ice cream into blender

Is milkshake too thick?

Get out ingredients

Add 1 cup milk

Blend all ingredients

Add more milk and blend ingredients again

Pour milkshake into glass

START

Drink your milkshake

PROGRAMMER'S PASTIME #10

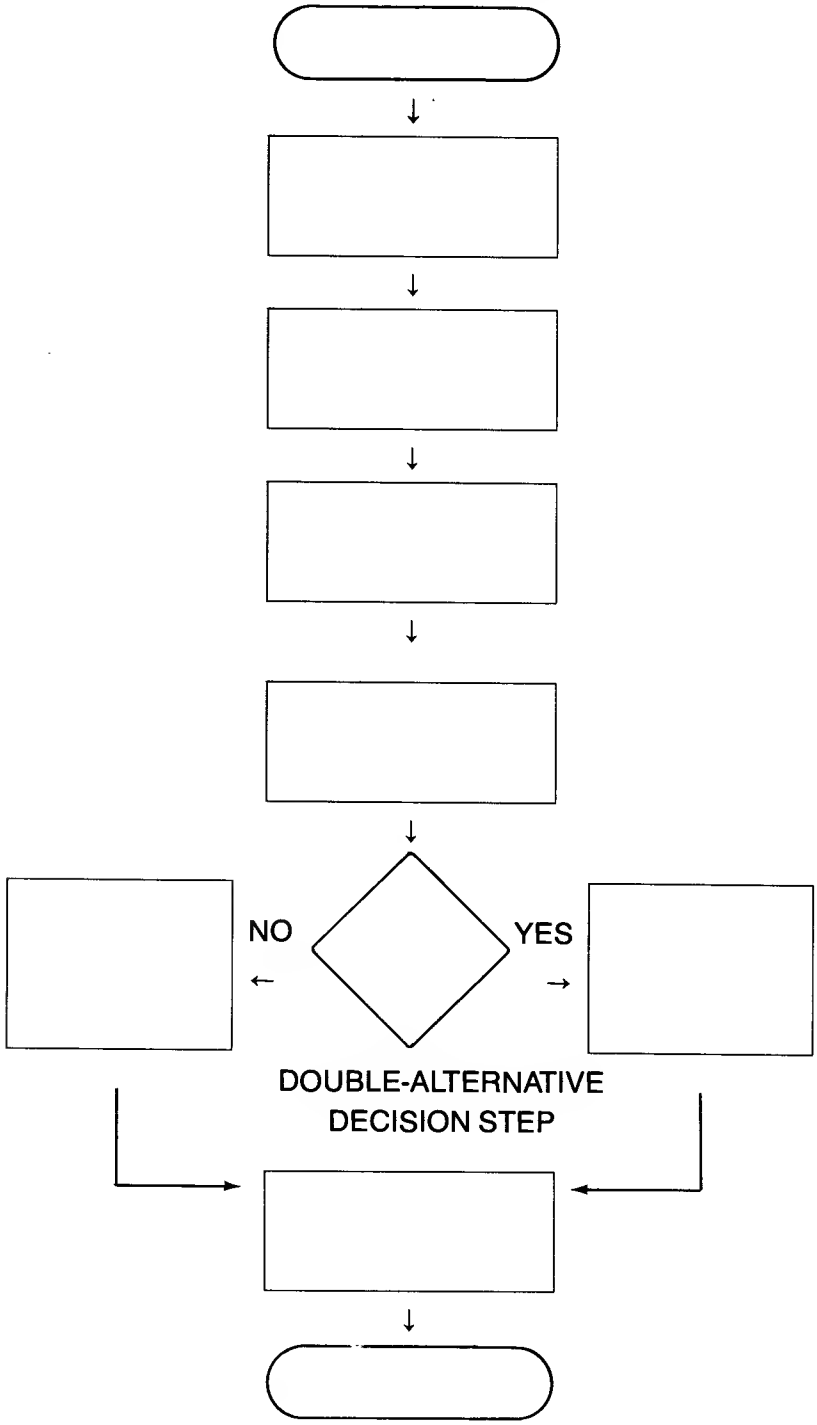
Write an algorithm on how to fix yourself a cold glass of water. Make the algorithm into a flow chart. Your flow chart should have a SINGLE-ALTERNATIVE DECISION STEP.

Ask the question, "IS THE WATER COLD ENOUGH?"
For the NO answer detour, your step might say:
ADD ICE CUBES.

PROGRAMMER'S PASTIME #11

For each flow chart, fill in the blank boxes with the steps you think would fit. Make sure your steps are in the right order.

Algorithm/Flow Chart: How to teach your pet bull to come when you call.



Missing Steps

Hold out handful of hay

Climb out of corral

START

Gently get bull's attention by calling his name

Pet bull. Quickly give him the hay

Say "COME TORO" over and over

Turn and run as fast as you can

STOP

Climb into corral. Approach bull carefully

Is the bull charging at you?

Algorithm/Flow Chart: How to bake cookies.

Missing Steps

Wait 10 min. then open oven

STOP

Are cookies done?

Gobble the cookies

Mix ingredients according to recipe

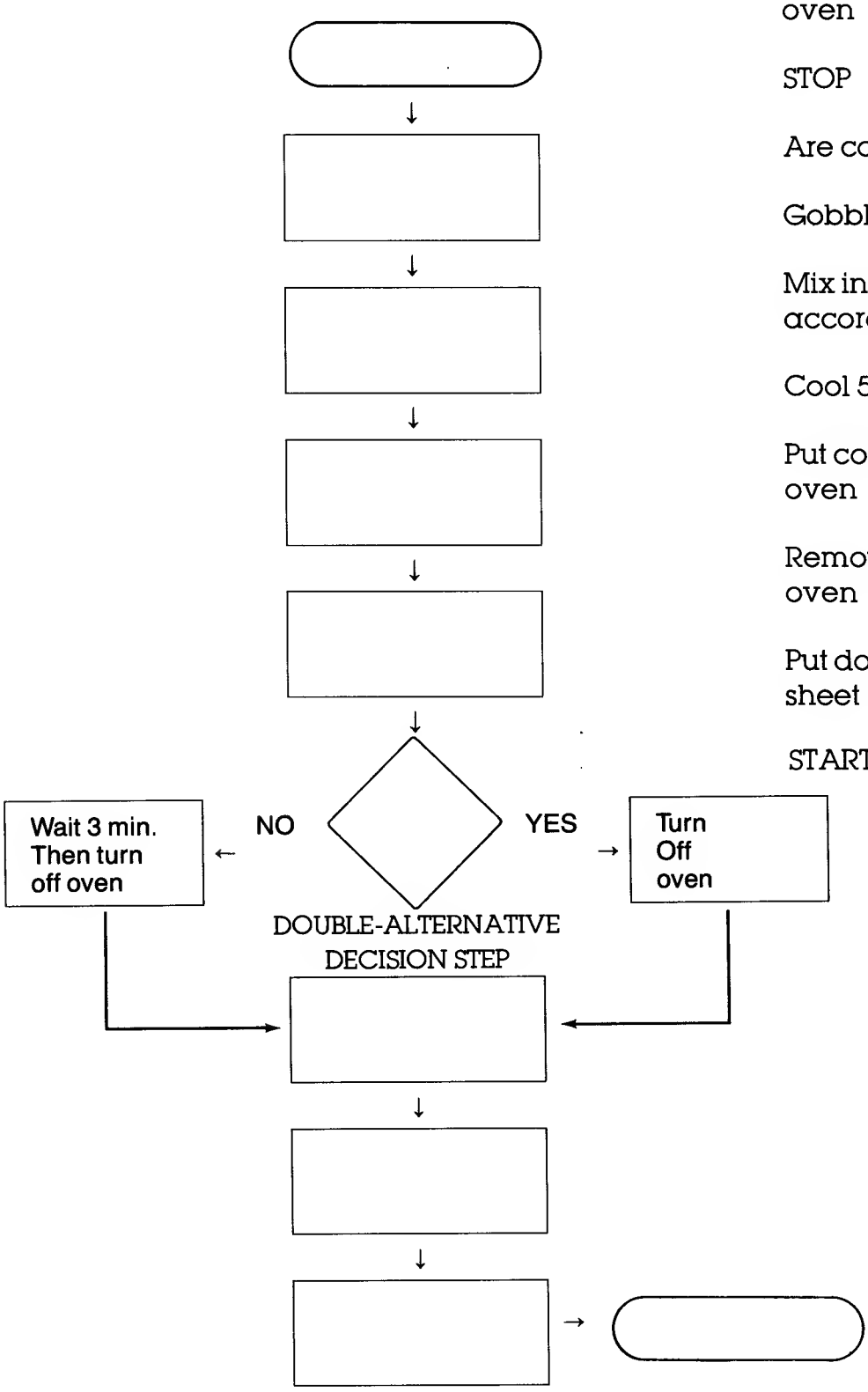
Cool 5 minutes

Put cookie sheets in hot oven

Remove cookies from oven

Put dough on cookie sheet

START



PROGRAMMER'S PASTIME #12

Write an algorithm and flow chart on how to hit a baseball with a bat during a game when you are up to bat. Your flow chart must have a DOUBLE-ALTERNATIVE DECISION STEP.

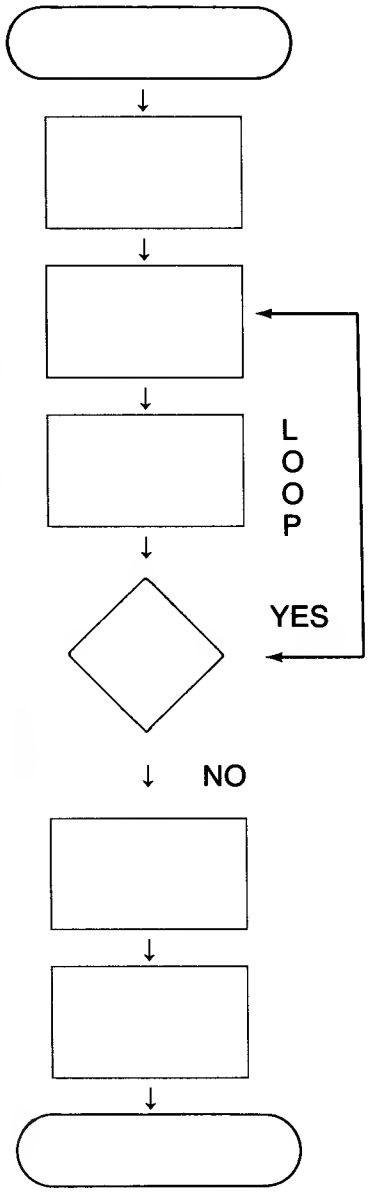
Question: IS THE BALL IN THE STRIKE ZONE?

YES	**NO**
KEEP YOUR EYE ON THE BALL	STEP AWAY FROM THE PLATE

PROGRAMMER'S PASTIME #13

Fill in the blank boxes of the flow chart with the step you think should fit. Make sure the steps fit the loop.

Algorithm/Flow Chart: How to eat candy.



Missing Steps

Is there more candy left?

Brush your teeth

STOP

Throw away wrapper

Unwrap candy

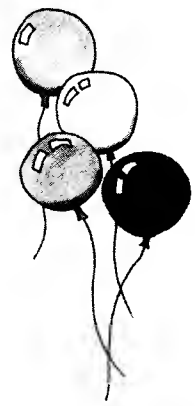
START

Chew and swallow

Take a bite

PROGRAMMER'S PASTIME #14

Write an algorithm for how to wash your hair. Write the algorithm in flow chart form. Your flow chart should have a LOOP.



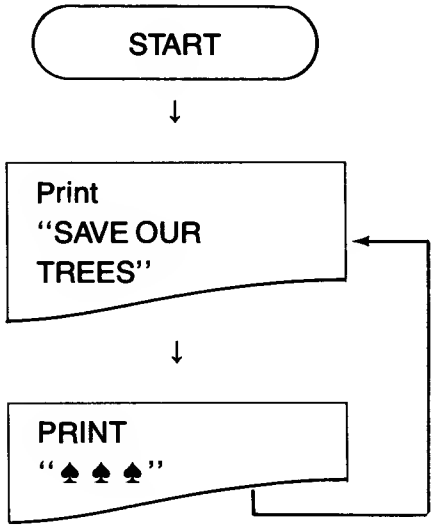
Example

Question: IS THERE SOAP IN YOUR HAIR?
LOOP: YES → RINSE YOUR HAIR

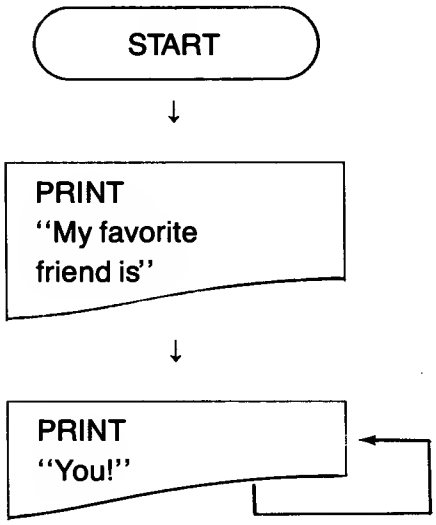
PROGRAMMER'S PASTIME #15

For each algorithm/flow chart, write a program in BASIC. (HINT — Pressing CTRL and [:] will give a tree-like graphic.)

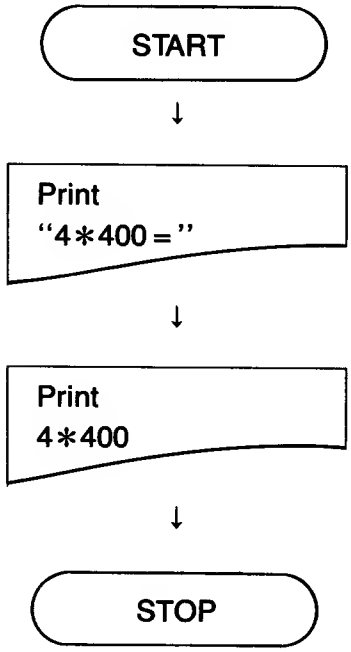
1.



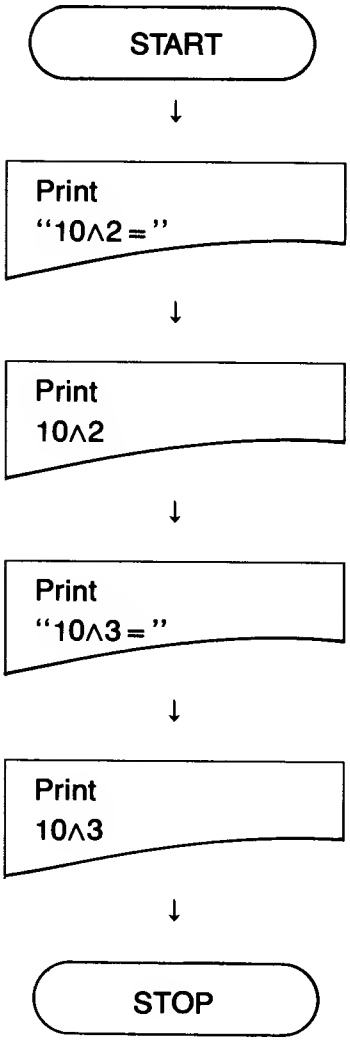
2.



3.


START
Print
"4*400 = "
Print
4*400
STOP

4.


START
Print
"10^2 = "
Print
10^2
Print
"10^3 = "
Print
10^3
STOP

PROGRAMMER'S PASTIME #16

Write an algorithm in a flow chart for each problem. Then write a BASIC program for ATARI to follow.

1. Tell ATARI to print
```
PETER PIPER
PICKED A PECK
OF PICKLED PEPPERS
```
over and over.

Flow chart **Program**

2. Tell ATARI to print
```
100*10-1000=
0
```

Flow chart **Program**

3. Tell ATARI to print

```
I CAN DO
56789*1234
WHICH EQUALS
70077626
```

Flow chart **Program**

4. Tell ATARI to print

```
SOMEWHERE
OVER THE RAINBOW
MANY COMPUTERS TRY
EQUATIONS LIKE
4*10=
(answer to 4*10)
```

Flow chart **Program**

PROGRAMMER'S PASTIME #17

For each program, write what you think ATARI would print.

Example:

```
10 ?"10+500+200="
20 ?10+500+200
30 GOTO 20
```

ATARI would print

```
10+500+200=
710
710
.  (The dots mean that
.  710 would be
.  printed forever.)
```

1.
```
10 ?"600-400+64="
20 ?600-400+64
30 END
```

ATARI would print

2.
```
10 ?"I LOVE      "
20 ?" YOU !!!    "
30 GOTO 20
```

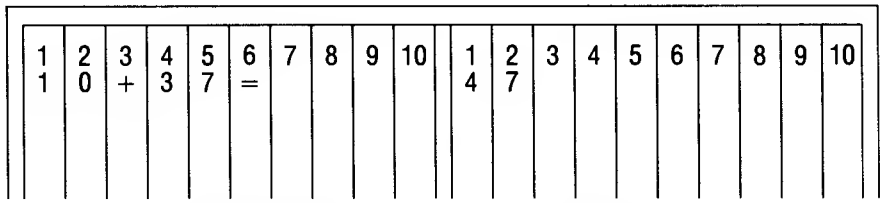
3.
```
10 ?"SHE SELLS"
20 ?"SEA SHELLS"
30 ?"BY THE SEASHORE"
40 ?"I CAN SAY IT ! "
50 GOTO 40
```

PROGRAMMER'S PASTIME #18

For each program, show how ATARI would print the equation on the screen.

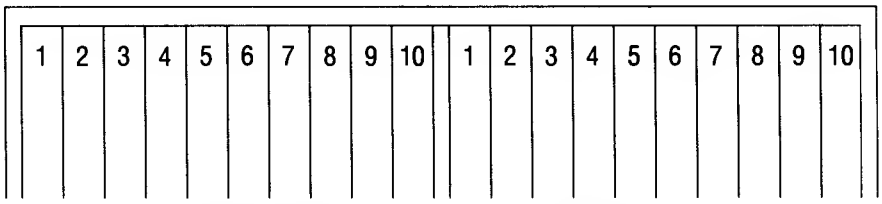
Example:

```
10 ?"10+37=",
   10+37
20 END
```



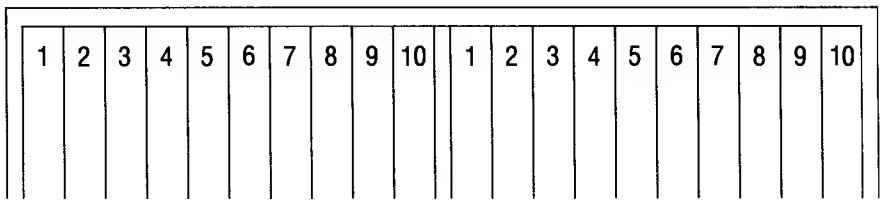
(1st Print Zone) (2nd Print Zone)

1.
```
10 ?"66+33=ø";
   66+33
20 END
```



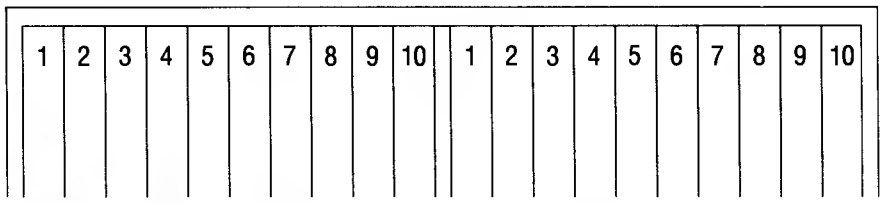
(1st Print Zone) (2nd Print Zone)

2.
```
10 ?"88-44+2=",
   88-44+2
20 END
```

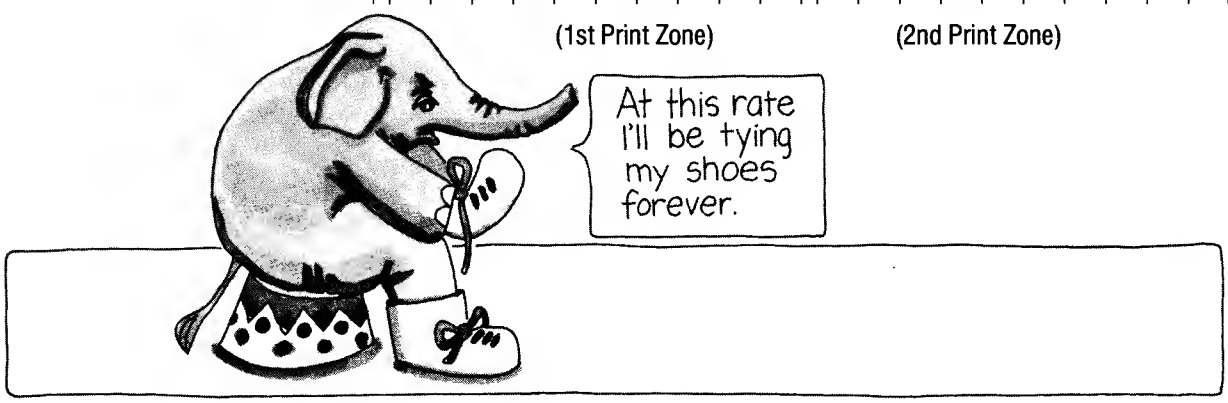


(1st Print Zone) (2nd Print Zone)

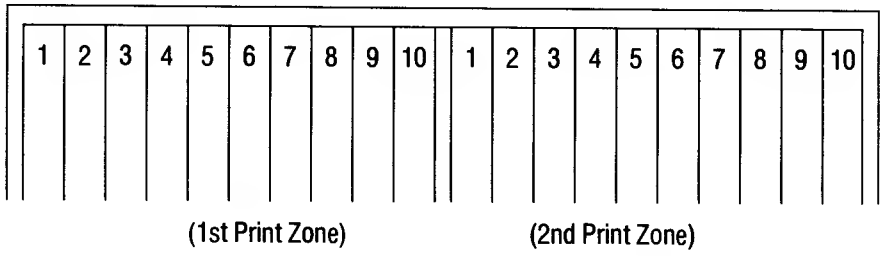
3.
```
10 ?"100+200+
   300=", 100+
   200+300
20 END
```



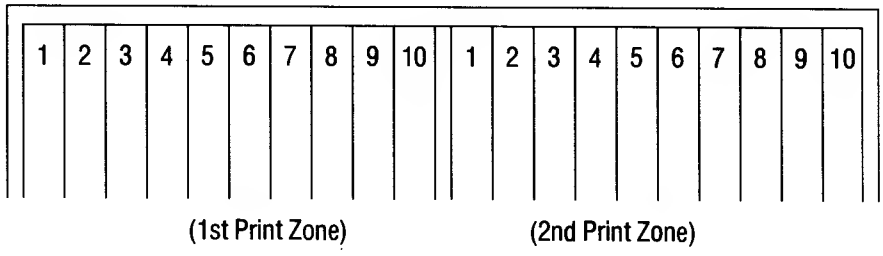
(1st Print Zone) (2nd Print Zone)



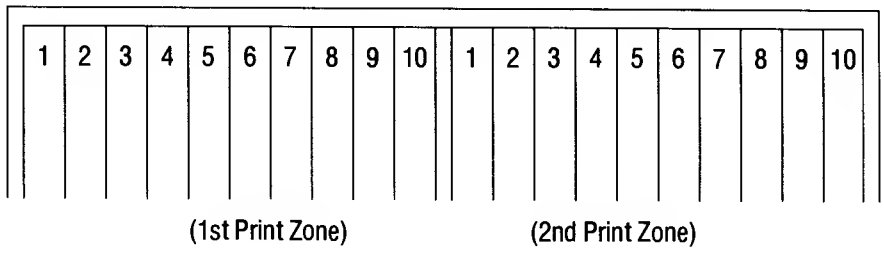
4.
```
10 ?"10*50=";
   10*50
20 ?
30 ?"10*50=",
   10*50
40 END
```



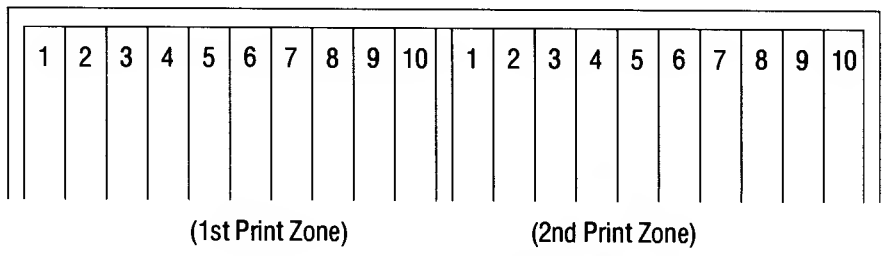
5.
```
10 ?"60/20=",
   60/20
20 ?
30 ?"60/20=";
   60/20
40 END
```

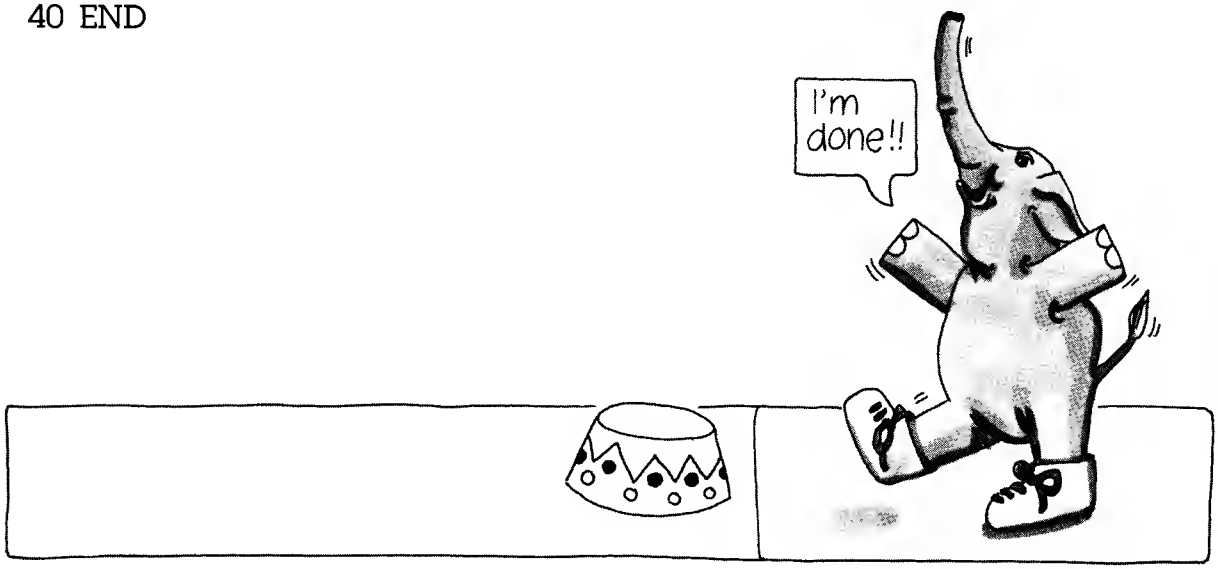


6.
```
10 ?"5-6=";
   5-6
20 END
```



7.
```
10 ?"8-7=";
   8-7
20 ?
30 ?"7-8=";
   7-8
40 END
```





Write a program that tells ATARI to print what is seen on the screens below.

Screens **Programs**

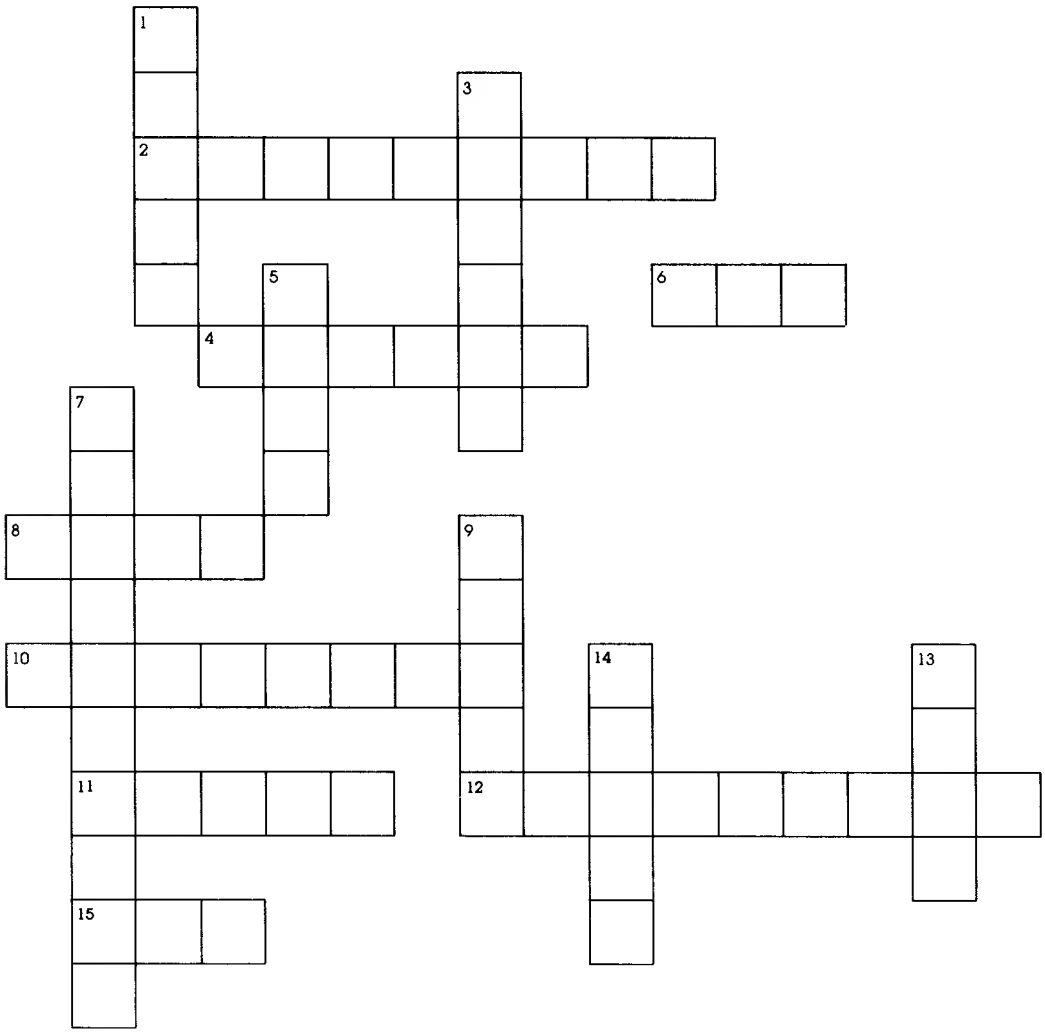
8.
```
ATARI IS A . . .  NUMBER CRUNCHER
```

9.
```
10-9=                    1
9-10=-1
```

10.
```
5-6=-1
6-5=                     1
```

11.
```
IF YOU CAN DO            8^8
YOU'RE A . . .
WHIZ KID
WHIZ KID
WHIZ KID
.
.
.
```

COMPONENT 3 FUN PAGE



Word Bank

TEN	NEW
COMMA	ALGORITHM
GOTO	STOP
DOUBLE	LOOP
ZONES	START
CHART	
PROCESSING	
SEMICOLON	
SINGLE	

Down

1. In the flow-diagramming process, we write our algorithm in a flow _____.
3. In a flow chart, one detour from a decision box is a _____-alternative decision step.
5. In a flow chart, a step that is repeated is called a _____.
7. The rectangle-shaped boxes in a flow chart that tell you to do something are called _____ boxes.
9. ATARI's screen has four print _____.
13. The last step in a flow chart is _____.
14. A _____ tells ATARI to go to the next print zone and then begin printing.

Across

2. An _____ is a step-by-step method that can be used to solve a problem.
4. Two detours from a decision box in a flow chart is a _____-alternative decision step.
6. Each print zone on ATARI's screen can hold _____ characters.
8. The BASIC command to loop is _____.
10. The diamond-shaped boxes in a flow chart that ask you questions are called _____ boxes.
11. The first step in a flow chart is _____.
12. A _____ tells ATARI to hold the cursor in place until the next PRINT command.
15. Each PRINT statement tells ATARI to print on a _____ line.

Evaluate Yourself

1. Component 3 was _____
   because _____
   _____
2. The best parts of the components were _____
   _____
   _____
3. The parts I liked the least were _____
   _____
   _____
4. The most valuable thing I learned in this component was _____
   _____
   because _____
   _____

PROGRAMMER'S PASTIME #19

For each LET statement, fill in the CONTENTS of the memory cell mailbox.

1. 10 LET OP=33

2. 20 LET T2=17

3. 30 LET M1=3000

4. 40 LET D=640

For each LET statement, fill in the ADDRESS and the CONTENTS of the memory cell mailbox.

1. 10 LET FF=99

2. 20 LET PQ=5

3. 30 LET N=0

4. 40 LET W7=62

Read each variable below. If it follows the rules we learned for writing variables, write "YES." If it does not follow the rules, write "NO."

1. PR	_____	7. X3	_____
2. ITC	_____	8. BB	_____
3. A	_____	9. 2CC	_____
4. E1	_____	10. 23D	_____
5. 3X	_____	11. FDd	_____
6. 7Z	_____	12. KK1	_____

PROGRAMMER'S PASTIME #20

Read each program. Then write what ATARI would print as the output. If you can, check your answers by running the programs on ATARI.

Program **Output**

1. 10 LET ZB=14
   20 ? ZB
   30 END

2. 10 LET T2=77
   20 ? "T2"
   30 END

3. 10 LET U=182
   20 ? "U"
   30 ? U
   40 END

4. 10 LET RC=7
   20 ? "RC IS",
   30 ? RC
   40 END

5. 10 LET GG4=66
   20 ? "GG4 ISØ";
   30 ? GG4
   40 END

PROGRAMMER'S PASTIME #21

Read each program. Then write what ATARI would print as the output. If you can, check your answers by running the programs on ATARI. Pay close attention to safe and unsafe variables!

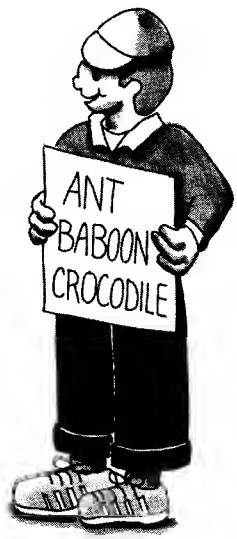
Program **Output**

1. 10 LET RB4=40
   20 LET RB5=50
   30 LET RB1=10
   40 ?RB5;"øISø";
   50 ?RB1;"øMORE";
   60 ?"THANø";RB4
   70 END

2. 10 LET T=5
   20 LET V=25
   30 ?"THE SQUARE
   ROOT OFø";
   40 ?V;"øISø";T
   50 END

3. 10 ?"MY FAVORITE
   NUMBER ISø";
   20 LET D=333
   30 ?D
   40 GOTO 30

4. 10 ?"MY FAVORITE
   NUMBER ISø";
   20 LET D=333
   30 ?D;
   40 GOTO 30



PROGRAMMER'S PASTIME #22

Read each program. Then write what ATARI would print as the output.

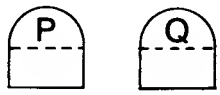
Program **Output**

```
1. 10 LET N=12
   20 LET R=6
   30 ?N/R
   40 END

2. 10 LET N=12
   20 LET R=6
   30 ?"N+R=";N+R
   40 END

3. 10 LET N=12
   20 LET R=6
   30 ?N;"+";R;"=";
      N+R
   40 END

4. 10 LET Q=10
   20 LET R=20
   30 LET S=30
   40 LET T=40
   50 ?T/R
   60 ?Q;"+";S;"=";
   T
   70 ?S-Q;"=";S;
      "-";Q
   80 END
```

PROGRAMMER'S PASTIME #23

For each LET statement, fill in the contents of the memory cell mailboxes.

1. 10 LET P=41
   20 LET Q=P



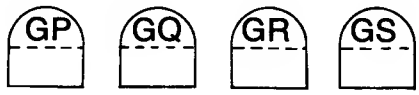
2. 10 LET A=36
   20 LET E=16
   30 LET I=A

3. 10 LET L=4
   20 LET M=2
   30 LET N=4+2

4. 10 LET B1=3
   20 LET B2=6
   30 LET B3=B1+B2

5. 10 LET AB=10
   20 LET AC=15
   30 LET AD=AC+5
   40 LET AE=AB+10

6. 10 LET GP=19
   20 LET GQ=GP-4
   30 LET GR=GQ+4
   40 LET GS=GR*1



PROGRAMMER'S PASTIME #24

Read each program. Then write what ATARI would print as the output. Check your answers by running the programs on ATARI.

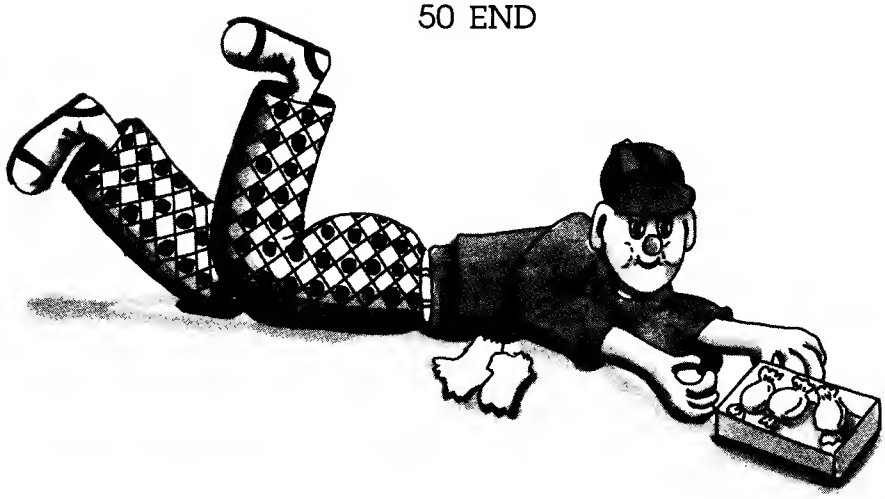
Program **Output**

1.
```
10 LET PJ=17
20 LET J2=34
30 LET J4=PJ+J2
40 ? J4
50 END
```

2.
```
10 LET B=2
20 ? B
30 LET B=100
40 ? B
50 END
```

3.
```
10 LET X1=2
20 LET X2=X1*5
30 LET X3=X2/X1
40 ? X3
50 END
```

4.
```
10 LET E6=3
20 LET E7=12
30 ? "PRODUCT",
   "QUOTIENT"
40 ? E6*E7, E7/E6
50 END
```



Program **Output**

```
5.10 LET HI=16
   20 LET HJ=HI+4
   30 ?HJ+10
   40 END

6.10 LET M=16
   20 LET N=14
   30 ?M+N
   40 LET N=12
   50 ?M+N
   60 END

7.10 LET Z1=8
   20 LET Z2=Z1-2
   30 ?Z2+Z1/2
   40 END

8.10 LET T1=6
   20 LET T1=7
   30 ?T1
   40 GOTO 30
   50 END

9.10 LET J=11
   20 LET K=22
   30 LET J=17
   40 ?K+J
   50 END
```

PROGRAMMER'S PASTIME #25

Read each program. Then write what ATARI would print as the output. Check your answers by running the programs on ATARI.

Program **Output**

1.
```
10 LET A=100
20 LET B=A/25
30 LET C=B*A
40 ?"B=";B
50 ?"C=";C
60 ?"IF A+B+C=";
   A+B+C
70 ?"THEN A=";A
80 END
```

2.
```
10 LET Q1=8
20 LET QZ=4
30 ?Q1/QZ
40 LET Q1=12
50 ?Q1/QZ
60 END
```

3.
```
10 LET C=9
20 LET D=8
30 LET C=7
40 ?C
50 ?D
60 END
```

PROGRAMMER'S PASTIME #26

In each program there are one or more mistakes. Find the mistake(s) and circle the line number where you found it. Then write the statement the correct way in the space to the right.

Program

Example:

```
10 LET 4=D
20 ?D
30 END
```

Correction

```
10 LET D=4
```

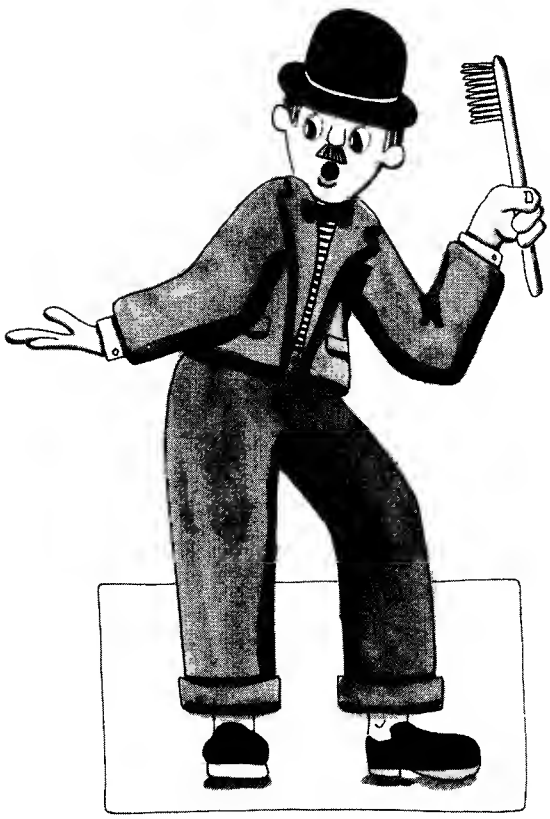
1.
```
10 LET L=44
20 LET 6=M
30 ?L+D
40 END
```

2.
```
10 ?Y
20 LET Y=66
30 END
```

3.
```
10 LET 5B=6
20 LET H=3
20 ?H
40 END
```

4.
```
10 ?"A+B"; A+B
20 LET A=3
30 LET B=4
40 END
```

Program	Correction
5. 10 LET XY=5 20 LET 3=VW 30 ?XY+VW 40 END	
6. 10 LET X=99 20 LET C=4 ?X−C 40 END	
7. 10 LET D+2=10 20 LET E=4 30 ?D−E 40 END	
8. 10 LET 3Y=2 20 LET B1=9 30 ?B1 40 END	



PROGRAMMER'S PASTIME #27

Using any shortcuts you have learned so far, rewrite the long programs below to make them as short as possible.

Program

Example:

```
10 LET B=49
20 LET E=409
30 PRINT B+E
40 END
```

Shortcut

```
10 LET B=49: LET E=409
20 ?B+E
30 END
```

1.
```
10 LET R=50
20 LET N=22
30 ?R-N
40 ?R+N
50 END
```

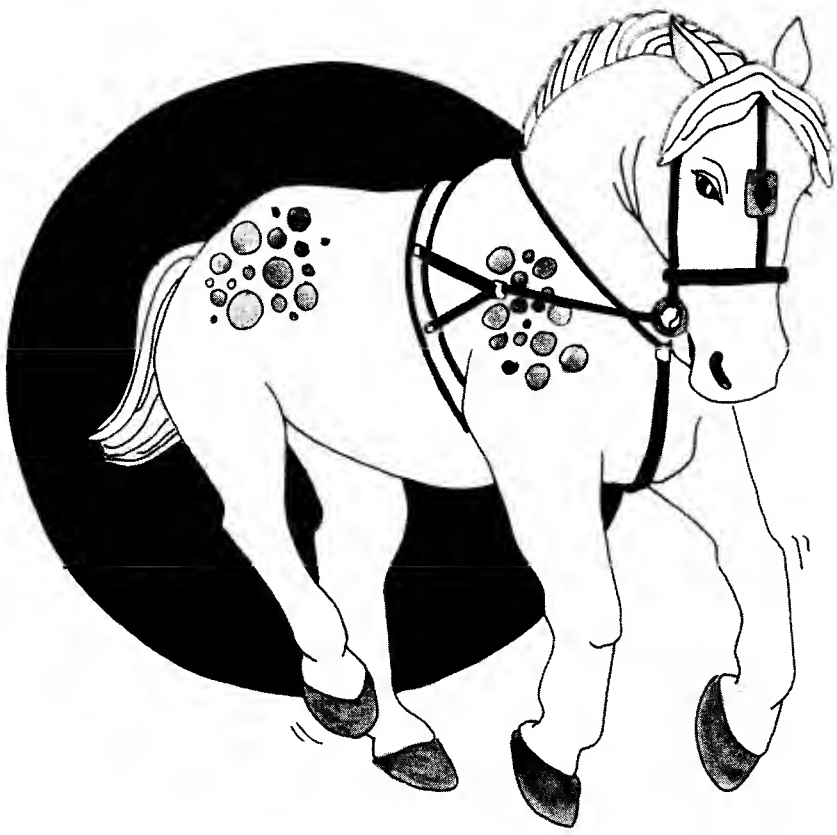
2.
```
10 LET S1=98
20 LET S2=89
30 LET S3=889
40 PRINT "S3-S2-S1=";
   S3-S2-S1
50 PRINT "S2*S1=";
   S2*S1
60 END
```

3.
```
10 LET U=40
20 LET T=20
30 ?U+T
40 ?U-T
50 ?U*T
60 ?U/T
70 END
```

Program **Shortcut**

4.
```
10 LET Y2=2
20 LET Y3=22
30 LET Y4=Y2+2
40 LET Y5=Y3+22
50 PRINT "Y4=";Y4
60 PRINT "Y5=";Y5
70 END
```

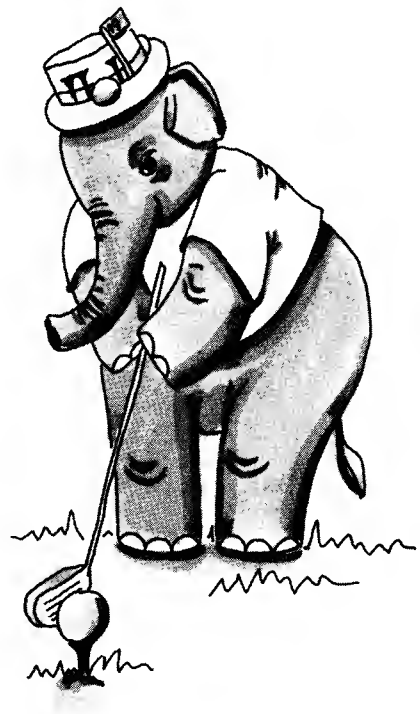
5.
```
10 LET D2=9
20 LET D4=18
30 ?"D2+D4=";
40 ?D2+D4
50 ?"D4-D2=";
60 ?D4-D2
70 END
```



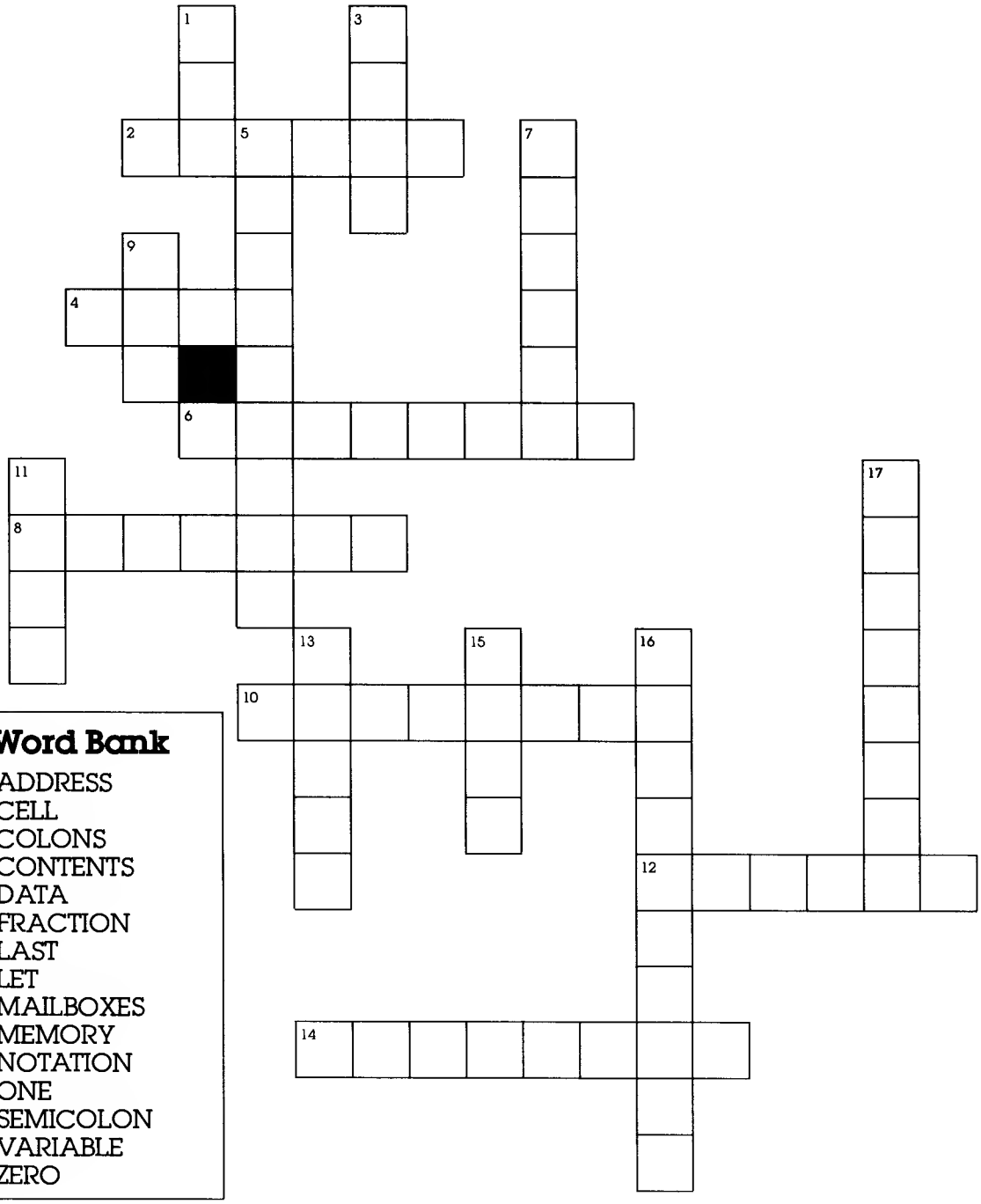
PROGRAMMER'S PASTIME #28

For each E notation number, write the whole number or decimal that it stands for.

E Notation	**Whole number or decimal**
Example: 7.982 E+07	79820000
1. 3.26 E+11	_____
2. 1.23 E−04	_____
3. 7.2 E+08	_____
4. 4.679 E−06	_____
5. 2.22 E+07	_____
6. 6.46982 E−03	_____
7. 5.3211 E+12	_____
8. 9.011 E−05	_____
9. 8.0045 E+13	_____
10. 1.467 E−07	_____
11. 8.006 E+09	_____
12. 6.002 E−03	_____
13. 3.9826 E−05	_____
14. 1.976345 E+08	_____



COMPONENT 4 FUN PAGE



Word Bank

ADDRESS
CELL
COLONS
CONTENTS
DATA
FRACTION
LAST
LET
MAILBOXES
MEMORY
NOTATION
ONE
SEMICOLON
VARIABLE
ZERO

Down

1. How many pieces of information can a memory cell store?
3. If ATARI sees a variable that has not been introduced by a LET statement, ATARI will automatically give that variable a value of _____.
5. ATARI's memory can be thought of as electronic _____.
7. Whatever ATARI prints is called _____.
9. The _____ statement assigns a value to a variable.
11. Information is also called _____.
13. In a LET statement, the variable must always come *before* the _____.
15. When you introduce the same variable more than once in a program, ATARI will remember the _____ value you gave it.
16. We can use commas and _____ to change the arrangement of the output in a program.
17. ATARI will change numbers with more than ten digits into E_____.

Across

2. Which part of ATARI's brain allows it to do many of the tricks we teach it?
4. A mailbox in ATARI's memory is also called a memory _____.
6. A memory cell consists of the address and the _____.
8. Each memory cell mailbox has its own _____.
10. The address of a memory cell is also called a _____.
12. We can use _____ to shorten our program in between LET statements.
14. The type of number ATARI can't understand is a _____.

Evaluate Yourself

1. Component 4 was _____
   because _____
   _____
2. The best parts of the component were_____
   _____
   _____
3. The parts I liked the least were_____
   _____
   _____
4. The most valuable thing I learned in this component was _____
   _____
   because_____
   _____
   _____

PROGRAMMER'S PASTIME #29

Read each program. Then write what you think ATRARI would print as the OUTPUT. Run the programs on ATARI to check your answers.

Program **Output**

1.
```
10 FOR Q=2 TO 6
20 ?Q
30 NEXT Q
40 END
```

2.
```
10 FOR Q=2 TO 4
20 ?"Q=";Q
30 NEXT Q
40 END
```

3.
```
10 FOR A=1 TO 5
20 ?"HELLO FRIEND!"
30 ?"HOW ARE YOU?"
40 NEXT A
50 END
```

4.
```
10 FOR D=1 TO 3
20 ?D
30 ?D+10
40 NEXT D
50 END
```

5.
```
10 LET P=3
20 FOR Q=1 TO 3
30 ?P;"+";Q;"=";P+Q
40 NEXT Q
50 END
```

Program **Output**

6.
```
10 FOR B=1 TO 5
20 ?"B","B+B","B*B"
30 ?B, B+B, B*B
40 NEXT B
50 END
```

7.
```
10 ?"MULTIPLICATION
   TABLE FOR 7"
20 FOR K=1 TO 12
30 ?K; "TIMES 7="; K*7
40 NEXT K
50 END
```

8.
```
10 FOR G=1 TO 10
20 ?"♡"
30 NEXT G
40 END
```

9.
```
10 FOR G=1 TO 10
20 ?"♡";
30 NEXT G
40 END
```

10.
```
10 FOR S=1 TO 10
20 LET S=S*S
30 ?S, S/S
40 NEXT S
50 END
```

Can you explain how this program works? ____

_____

_____

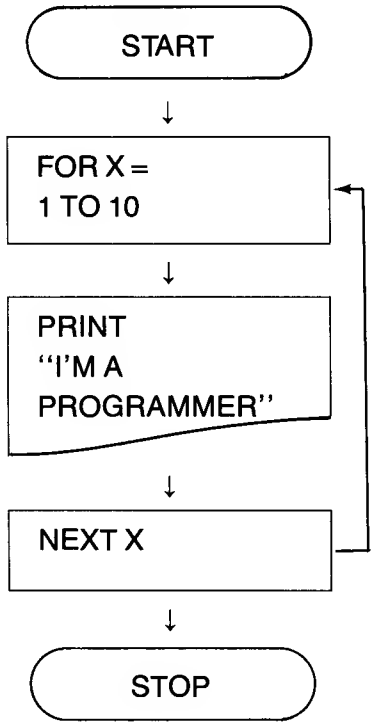
_____

PROGRAMMER'S PASTIME #30

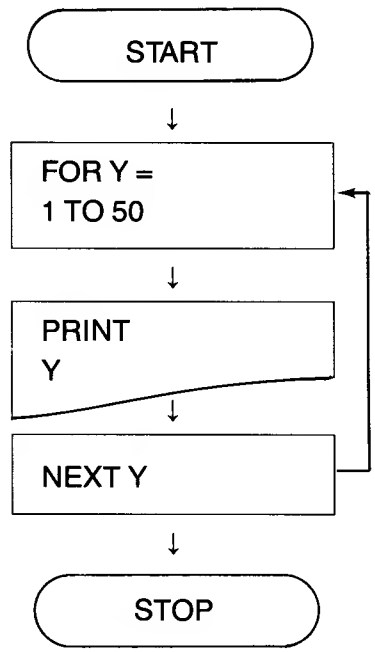
Write a program for each flow chart. Be sure to use a FOR–NEXT loop. Run your programs on ATARI to make sure they work.

Flow chart **Program**

1.

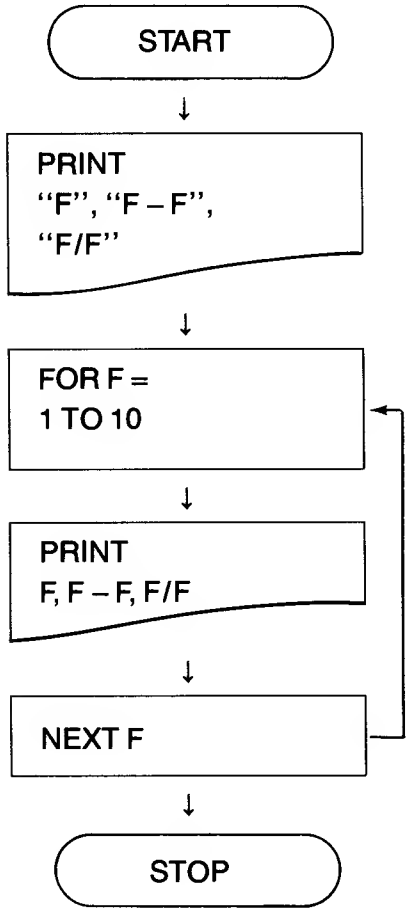


2.



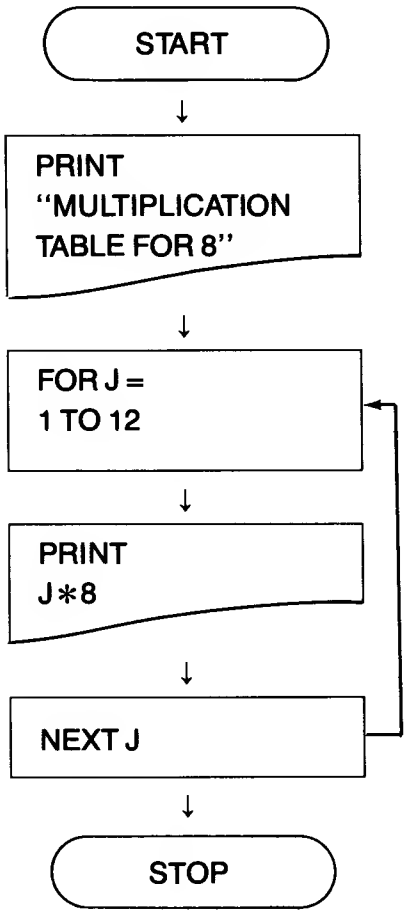
Flow chart **Program**

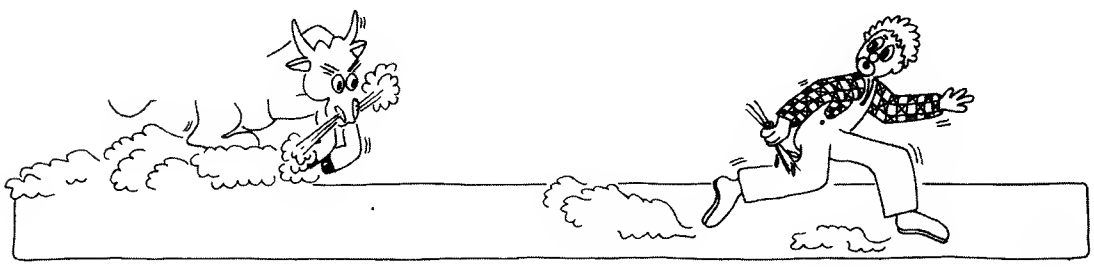
3.



Flow chart **Program**

4.





For each program description, write an algorithm in flow chart form. Then write the program. Run each program on ATARI to make sure it works.

Description	Flow chart	Program
5. Add something new to the program in #4 so it prints: J; "TIMES 8="; J*8 each time the loop is done.		
6. Write a program that prints the numbers from 1 to 20, their squares (X*X), and their cubes (X*X*X).		

Description	Flow chart	Program
7. Write a program that prints * 20 times on one line.		
8. Write a program that introduces D = 5 and P = 1 to 5. Make the program add D plus each value of P, and print the sums of D+P.		

PROGRAMMER'S PASTIME #31

In each program there is one mistake. Find the mistake and circle the line number where you found it. Then write the statement the correct way in the space to the right.

Program **Correction**

1. 10 FOR P=1−40
   20 ?P
   30 NEXT P
   40 END

2. 10 FOR W IS 6 TO 30
   20 ? W
   30 NEXT W
   40 END

3. 10 FOR E=3 TO 10
   20 ?E
   30 E NEXT
   40 END

4. 10 FOR L=1 TO 4
   20 ?L
   30 ?L*2
   40
   50 END

5. 10 LET G=4
   20 FOR H=5 TO 9
   30 ?G+H
   40 NEXT G
   50 END

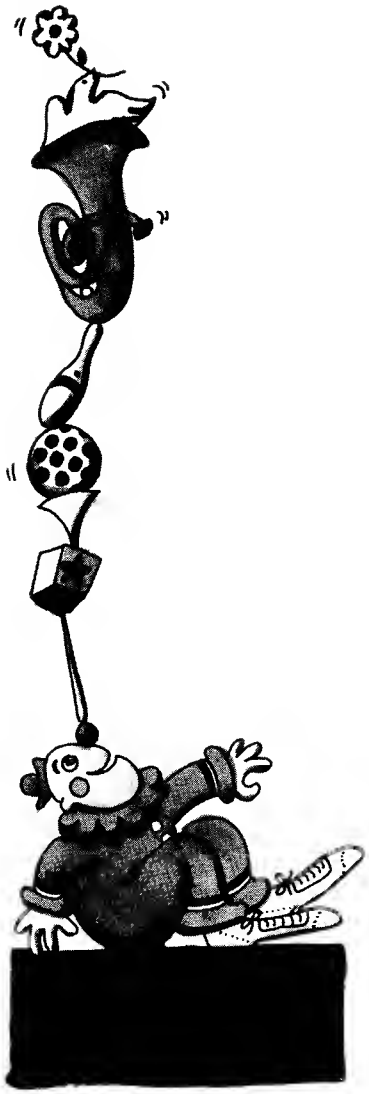
PROGRAMMER'S PASTIME #32

Read each program. Write what you think ATARI would print as the OUTPUT. Run the programs on ATARI to check your answers.

Program **Output**

1.
```
10 FOR F=0 TO 8
   STEP 2
20 ? F
30 NEXT F
40 END
```

2.
```
10 FOR J=18 TO 0
   STEP -3
20 ? J
30 NEXT J
40 END
```



3.
```
10 FOR B2=3 TO 21
   STEP 3
20 ? "HOWDY"
30 NEXT B2
40 END
```

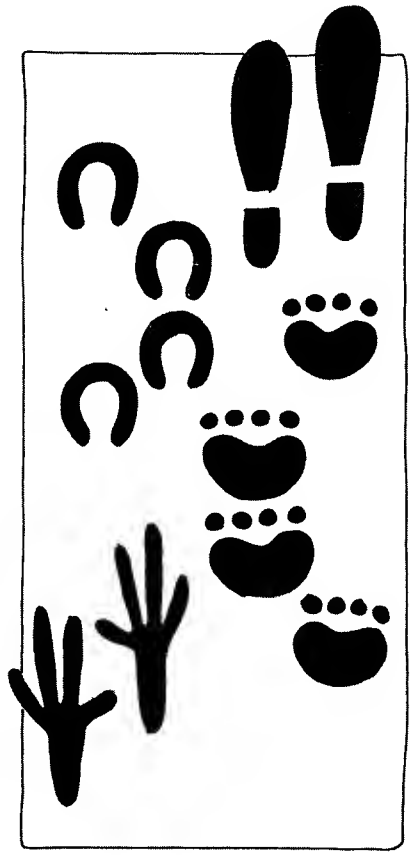
4.
```
10 LET N=5
20 FOR T=1 TO N
30 ? T
40 NEXT T
50 END
```

5.
```
10 LET M2=10
20 FOR S=0 TO 12
   STEP M2/5
30 ? S
40 NEXT S
50 END
```

Program

```
6.10 ?''IF SEPT 1 IS
     A SUNDAY THEN''
  20 FOR JJ=1 TO 31
     STEP 7
  30 ?''SEPTØ''; JJ;
     ''ØIS A SUNDAY''
  40 NEXT JJ
  50 END

7.10 LET PX=8
  20 FOR A7=0 TO 10
     STEP PX/4
  30 ? A7
  40 NEXT A7
  50 END

8.10 FOR ZZ=1 TO 14
     STEP 4
  20 ? ZZ
  30 NEXT ZZ
  40 END

9.10 FOR BD=20 TO 2
     STEP -5
  20 ? BD
  30 NEXT BD
  40 END
```

Output



PROGRAMMER'S PASTIME #33

For each program description, write an algorithm in flow chart form. Then write the program. Run each program on ATARI to make sure it works.

Description	Flow chart	Program
1. Write a program that tells ATARI to count from 0 to 40 by fours.		
2. Write a program that tells ATARI to count backwards from 8 to 0.		

Description	Flow chart	Program
3. Write a program that tells ATARI to print "HELLO" five times. Use a STEP statement.		
4. Write a program that tells ATARI to print the numbers 0 through 21 STEP N/4. Make N=12.		

Description	Flow chart	Program
5. Write a program that tells ATARI to print the numbers 1, 4, 7, 10 and 13. Use a STEP statement.		

PROGRAMMER'S PASTIME #34

Study each program. Write what you think ATARI would print as the OUTPUT. Run each program to check your answers.

Program **Output**

1.
```
10 LET C=0
20 FOR FL=1 TO 100
   STEP 10
30 ?"THINK"
40 LET C=C+1
50 ?C
60 NEXT FL
70 END
```

2.
```
10 LET C=0
20 ?"BRAIN POWER"
30 LET C=C+1
40 ?C
50 GOTO 20
```

3.
```
10 LET C=0
20 FOR FL=1 TO 8
30 LET C=C+1
40 ?C
50 ?"AWESOME"
60 NEXT FL
70 END
```

Program **Output**

4.
```
10 LET C=0
20 FOR Z=1 TO 50 STEP 10
30 ?"RAINBOW"
40 LET C=C+1
50 NEXT Z
60 ?"I PRINTED"
70 ?"RAINBOW"
80 ?C;"TIMES"
90 END
```

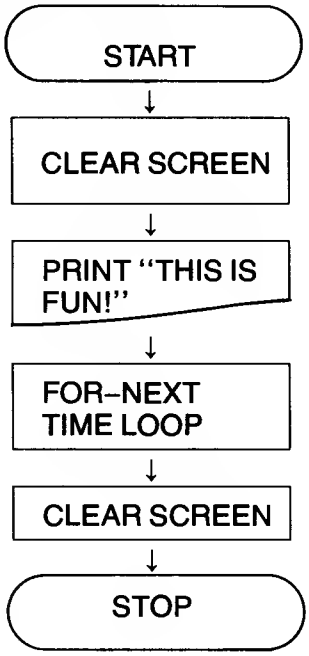
5.
```
10 LET C=0
20 FOR FL=1 TO 4
30 LET C=C+1
40 ?C
50 ?"JELLY BEANS"
60 NEXT FL
70 ?"A TOTAL OFØ";C;"ØJELLY
   BEANS"
80 END
```

PROGRAMMER'S PASTIME #35

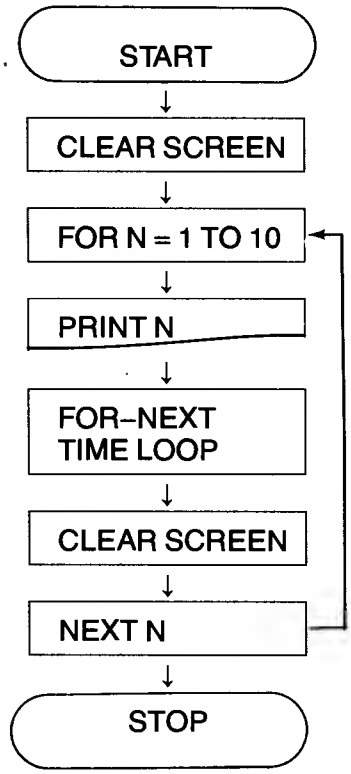
Write a program for each flow chart, then run the programs.

Flow Chart **Program**

1.

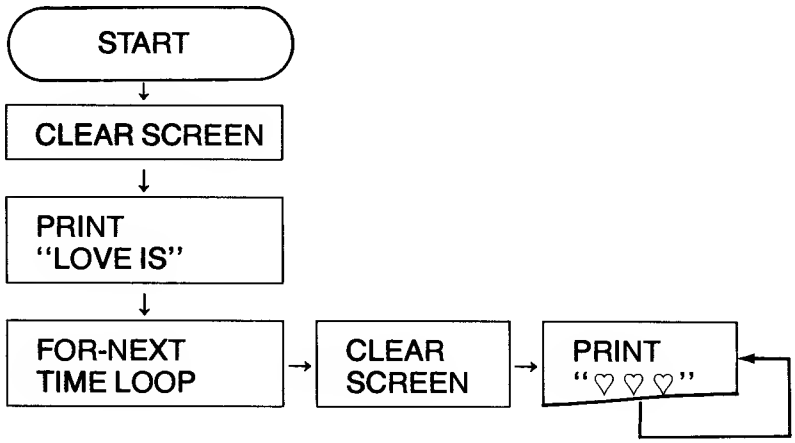


2.

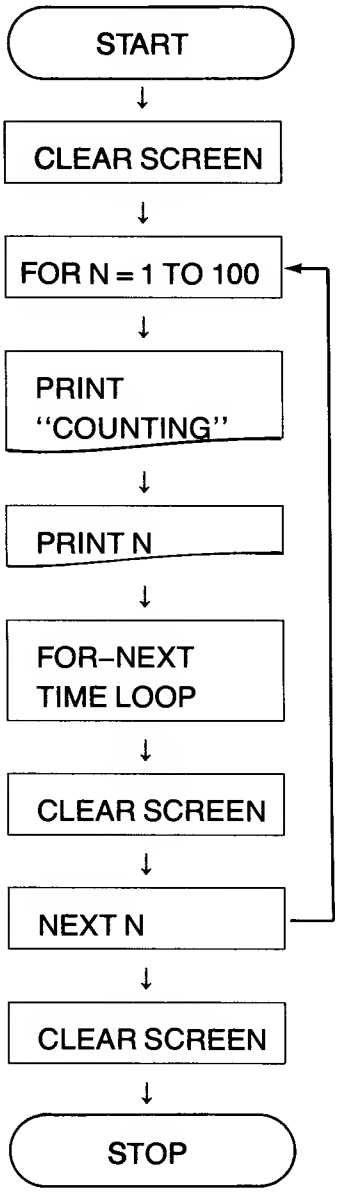


Flow Chart **Program**

3.

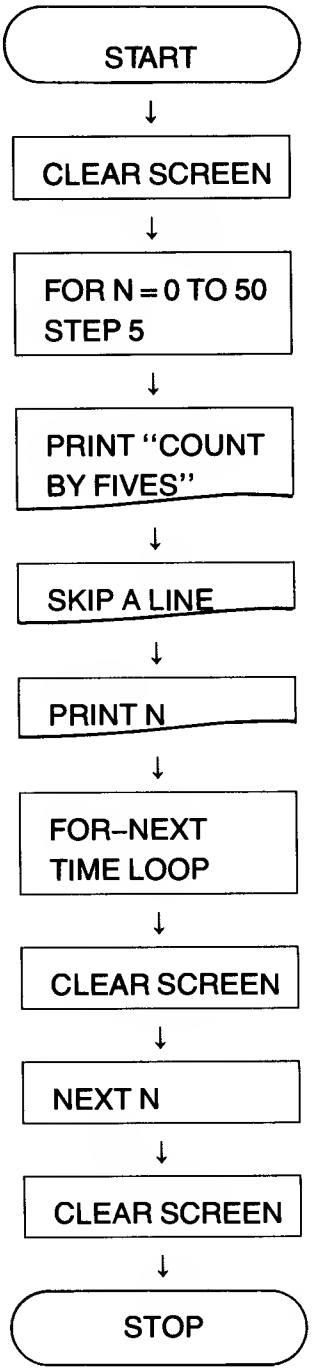


4.



Flow Chart **Program**

5.



PROGRAMMER'S PASTIME #36

You have learned how to program ATARI to move down a number of lines on the screen and then begin printing. To do this, you used a statement like this:

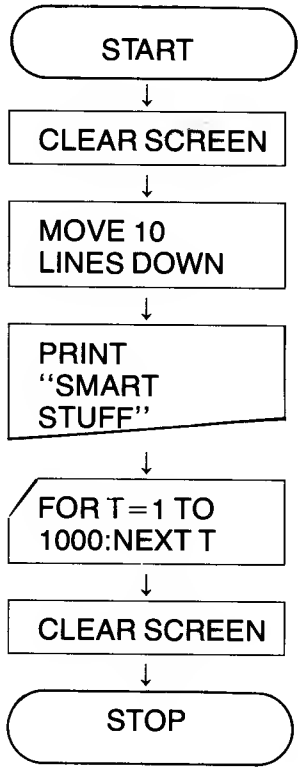
```
20 ?:?:?:?
```

This tells ATARI to move down 4 lines.

Now using the tricks you have learned to make ATARI clear the screen, and move the writing down several lines, write a program for the following flow charts. Try the programs out on ATARI.

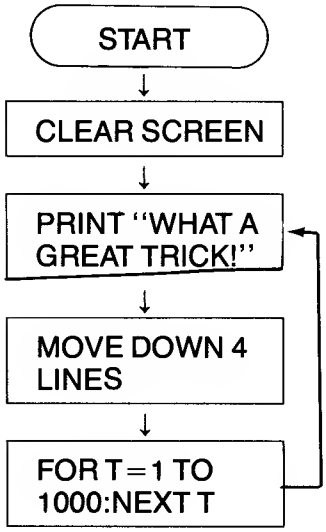
Flow Chart **Program**

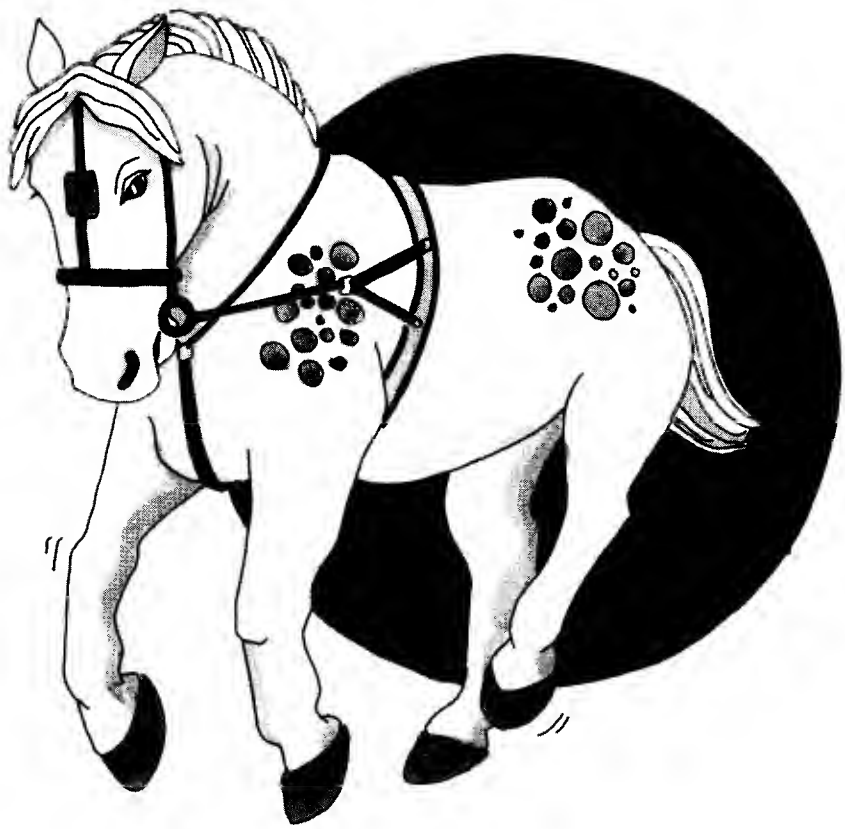
1.



Flow Chart **Program**

2.



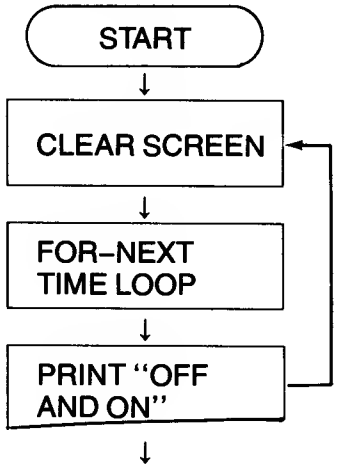


PROGRAMMER'S PASTIME #37

Write a program for each flow chart, then run your programs on ATARI to make sure they work.

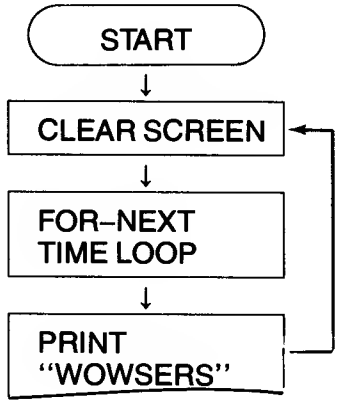
Flow Chart **Program**

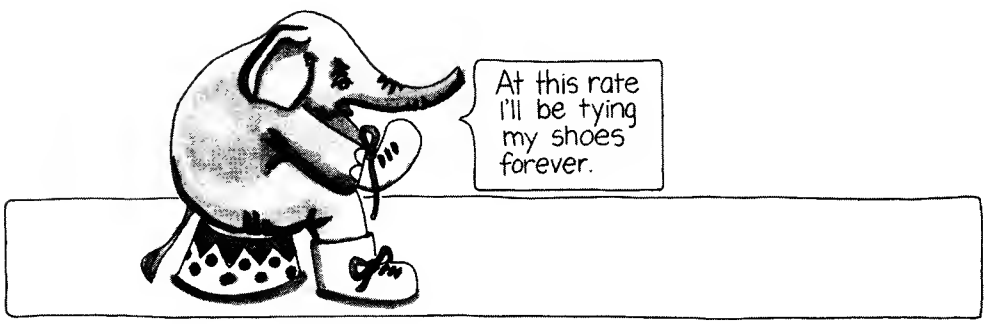
1.



This is the basic algorithm for making something blink.

2.

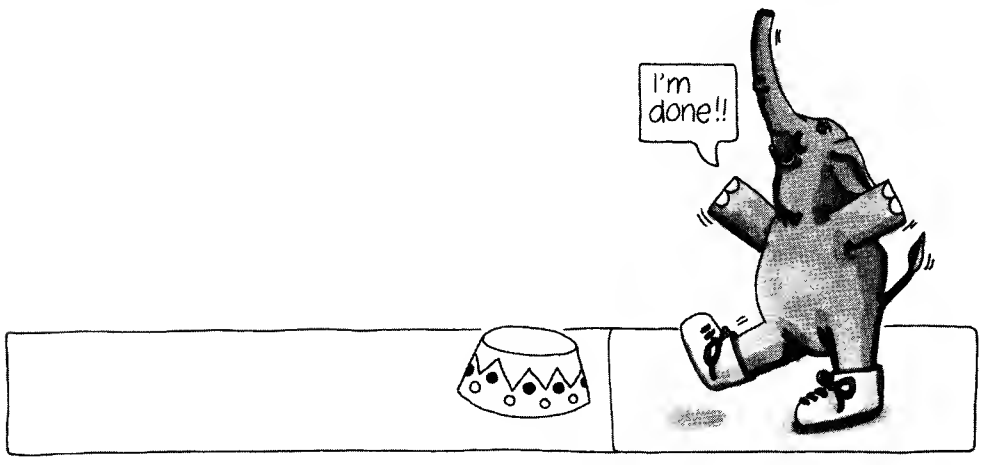


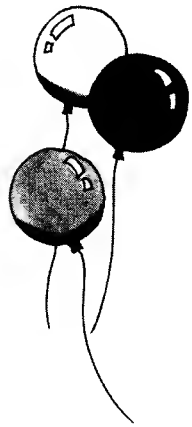


Use your expertise and your imagination to write two of your own programs that make something blink. You can even make graphics or pictures blink! Don't be afraid to experiment.

Flow Chart **Program**

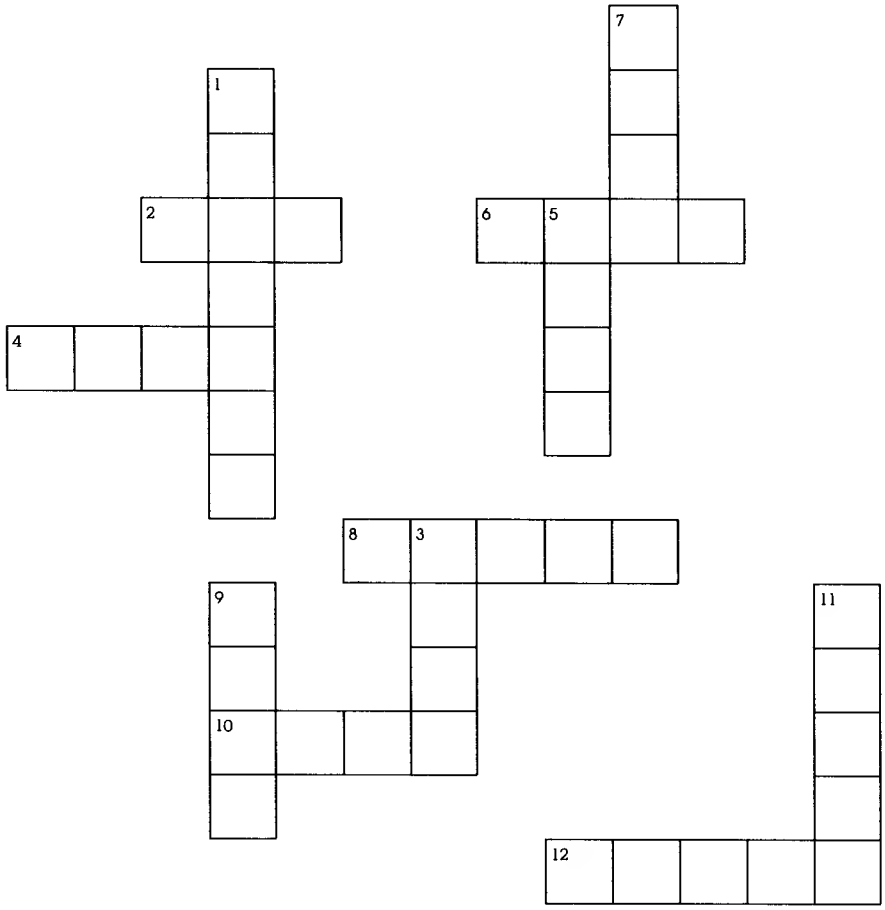
1.





2.

COMPONENT 5 FUN PAGE



Word Bank

BLINK	LOOP
BREAK	NEXT
BUG	SAVE
CLEAR	STEP
COUNTER	STOP
LIST	TIME

Down

1. A FOR–NEXT loop creates _____ controlled loops in a program.
3. A counter also lets you keep track of how many times you have done a _____.
5. To slow down ATARI's printing, use a FOR–NEXT _____ loop.
7. When you copy a program onto a cassette tape, you use the _____ command.
9. To see how a program is written, type _____.
11. We use the FOR–NEXT time loop to make things _____.

Across

2. A mistake in a program is called a _____.
4. Every FOR statement must have a _____ statement after it somewhere in the program.
6. We tell ATARI to count in a certain pattern by using the _____ command.
8. The statement ? " [ESC] [SHIFT] [CLEAR <] " tells ATARI to _____ the screen.
10. We push [BREAK] to make a program _____ running.
12. We _____ out of a program when we stop it before it has finished running.

Evaluate Yourself

1. Component 5 was _____
   because _____
   _____
2. The best parts of the component were _____
   _____
   _____
3. The parts I liked the least were _____
   _____
   _____
4. The most valuable thing I learned in this component was _____
   because _____
   _____

Other comments:

PROGRAMMER'S PASTIME #38

Study each program and write what you think ATARI would print as the output. Run the programs to check your answers.

Program **Output**

1.
```
 5 DIM F$(10)
10 LET F$="44"
20 LET F=44
30 ?F$ ; ?F
40 END
```

2.
```
 5 DIM G$(10)
10 LET G$="6+32="
20 LET G=6+32
30 ?G$ ; G
40 END
```

3.
```
 5 DIM A$(10), S$(12)
10 LET A$="ADDITION"
20 LET S$="SUBTRACTION"
30 ?A$ , S$
40 LET A=8+8
50 LET S=8-8
60 ?"8+8="; A, "8-8="; S
70 END
```

4.
```
 5 DIM Q$(10), R$(10)
10 LET Q$="HI HO"
20 LET R$="SILVER!"
30 ? Q$, R$
40 GOTO 30
```

5.
```
 5 DIM Z$(20)
10 LET Z$="YOU'RE OUTA SIGHT!"
20 FOR C=1 TO 20
30 ?Z$
40 NEXT C
50 END
```

PROGRAMMER'S PASTIME #39

There is at least one mistake in each program. Find the mistake(s), circle the line number where you found it, then write the statement(s) the correct way in the space to the right.

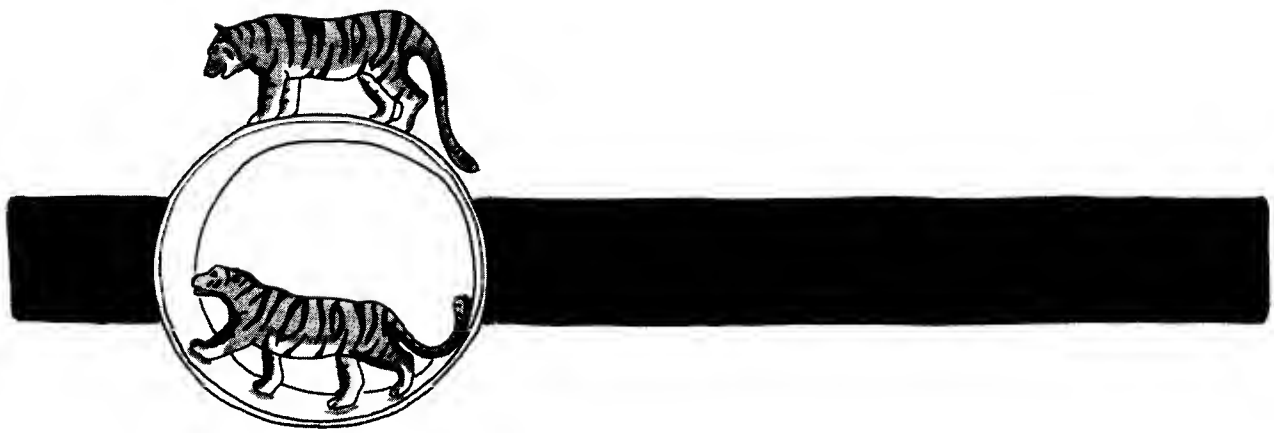
Program **Correction**

1.
```
10 LET AZ$="YES"
20 LET BY$="NO"
30 ? AZ$, BY$
40 END
```

2.
```
 5 DIM T$(10), U$(10)
10 LET T$="THE TIME"
20 LET U$="IS NOW"
30 ?T, U
40 END
```

3.
```
 5 DIM J$(10), K$(10)
10 LET J$= UP, UP
20 LET K$= AND AWAY
30 ? J$ : ? K$
40 END
```

4.
```
 5 DIM P$(20), T$(20)
10 LET"PARTRIDGE IN"=P$
20 LET"A PEAR TREE"=T$
30 ?P$, T$
40 END
```



PROGRAMMER'S PASTIME #40

Study each program. Write what you think ATARI would print as the output. You may write what you would answer for each INPUT statement. Run the programs to check your work.

Program **Output**

1.
```
 5 DIM A$(20)
10 ?"HOW OLD ARE YOU";
20 INPUT A
30 ?"WHAT'S IT LIKE TO BE";A;
40 INPUT A$
50 ?"I'M GLAD TO HEAR IT'S";A$
60 END
```

2.
```
 5 DIM A$(20)
10 ?"HOW OLD ARE YOU?"
20 INPUT A
30 ?"WHAT'S IT LIKE TO BE"; A; "?"
40 INPUT A$
50 ?"SO YOU ARE";A$;"TODAY"
60 END
```

3.
```
10 ?"HOW MANY BROTHERS AND"
20 ?"SISTERS DO YOU HAVE?"
30 ?"TYPE NUMBER OF BROTHERS,"
40 ?"A COMMA,"
50 ?"AND NUMBER OF SISTERS"
60 INPUT B, S
70 LET T=B+S
80 ?"SO YOU HAVE";T;"SIBLINGS"
90 END
```

Program **Output**

4.
```
10 ?"CHOOSE TWO NUMBERS
   AND"
20 ?"I WILL ADD THEM FOR YOU"
30 ?"TYPE 1ST NUMBER, COMMA,"
40 ?"THEN TYPE THE 2ND NUMBER"
50 INPUT F, S
60 ?
70 ?F;"+";S;"=";F+S
80 ?
90 ?"I'M A WHIZ!"
100 END
```

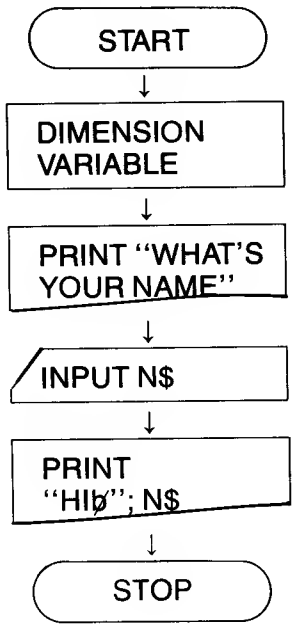
5.
```
10 ?"TYPE IN 2 NUMBERS"
20 ?"SEPARATED BY A COMMA"
30 INPUT O, T
40 ?
50 ?O;"+";T;"=";O+T
60 ?O;"-";T;"=";O-T
70 ?O;"*";T;"=";O*T
80 ?O;"/";T;"=";O/T
90 ?"SEE . . .I TOLD YOU"
100 ?"I WAS A WHIZ!"
110 END
```

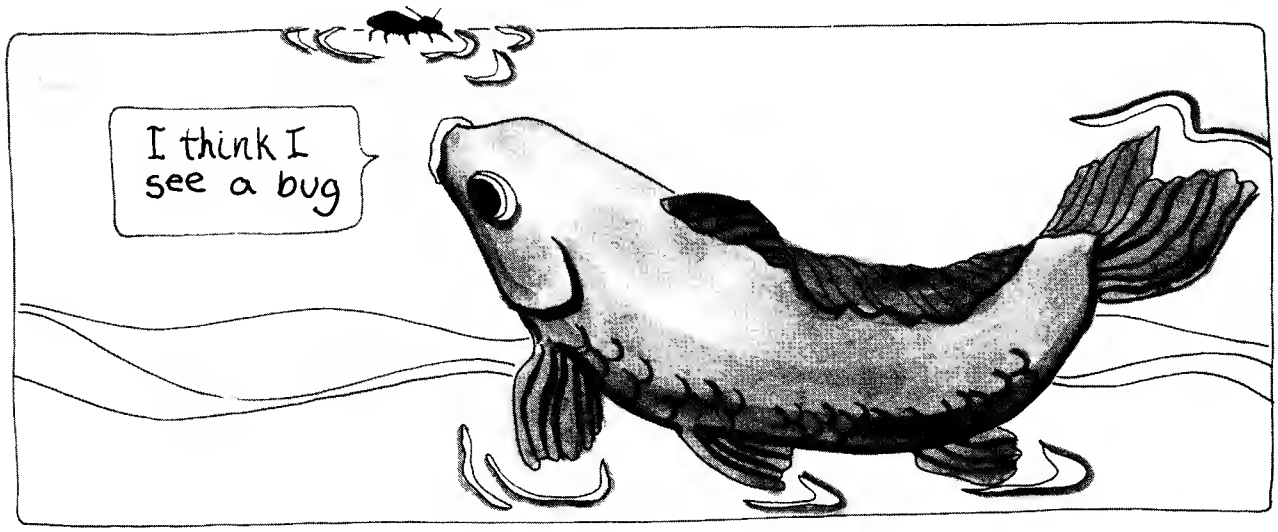
PROGRAMMER'S PASTIME #41

Write a program for each flow chart. Run your programs on ATARI to check for bugs.

Flow Chart **Program**

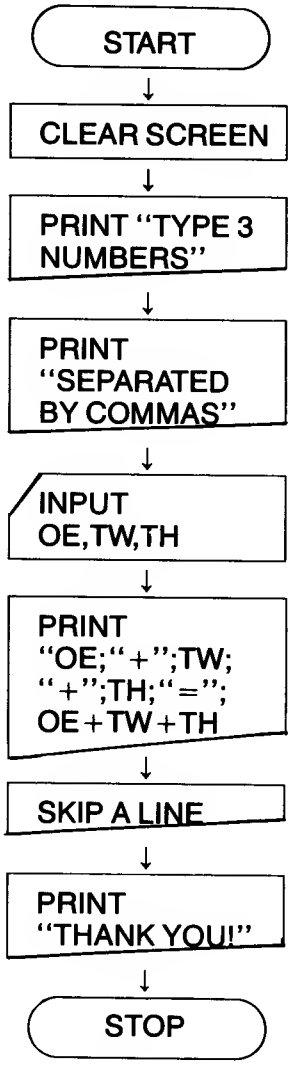
1.





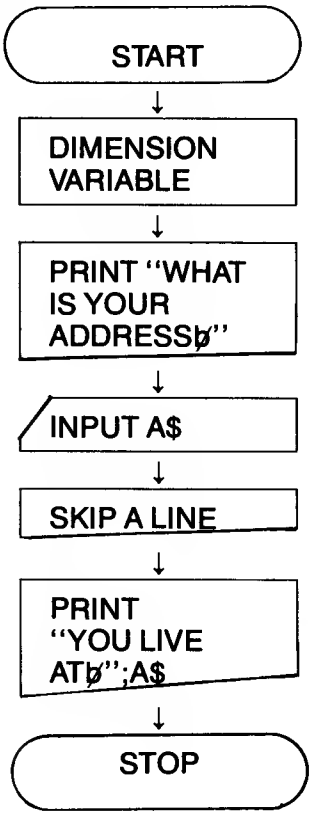
Flow Chart **Program**

2.



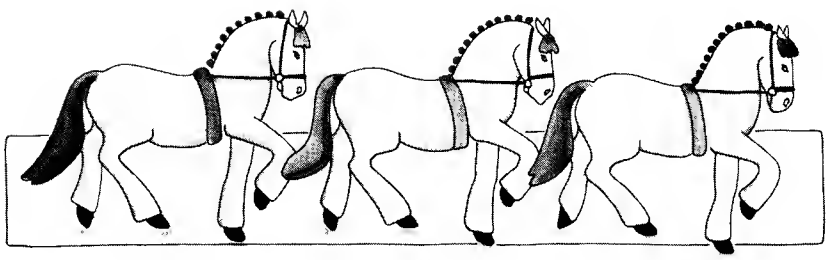
Flow Chart **Program**

3.



Write 3 programs using the INPUT statement. Write the flow chart for the algorithm first, then write the program. Debug your programs by running them on ATARI.

Flow Chart **Program**



PROGRAMMER'S PASTIME #42

Write each equation as an IF–THEN statement.

Question

Example:

Is A equal to C?

1. Is L$ equal to "MAYBE"?
2. Is F1 not equal to FZ?
3. Is GH greater than HI?
4. Is S$ less than or equal to F$?
5. Is X times B less than P times Q?
6. Is T divided by W greater than or equal to W times B?
7. Is P$ greater than M$?
8. Is the square root of Y equal to D?
9. Is G$ not equal to "NO"?
10. Is 10 divided by 5 less than 14 divided by 2?
11. Is Y$ equal to the square root of 64?
12. Is A plus B greater than D$?

IF–THEN statement

IF A=C THEN _____

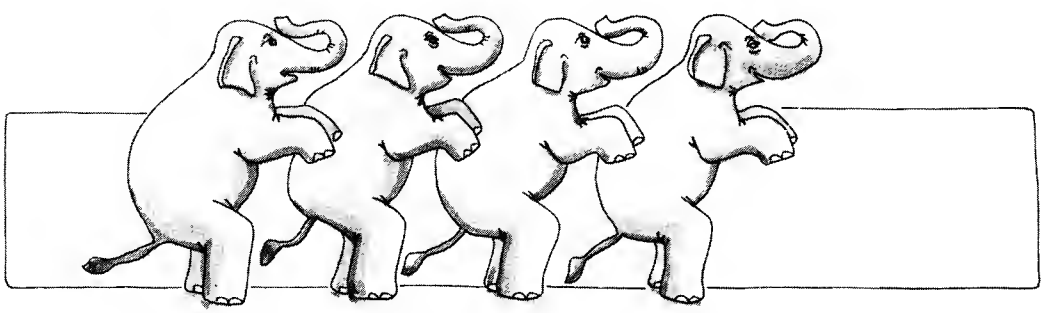
PROGRAMMER'S PASTIME #43

For each question write the Complement IF–THEN statement.

Question
Example
Is P$ equal to ''YES''?

Complement
IF–THEN statement
IF P$ < > ''YES'' THEN _____

1. Is QR greater than 2? _____
2. Is Z$ not equal to ''END''? _____
3. Is F less than or equal to P? _____
4. Is G$ equal to ''JEEPERS''? _____
5. Is S1 greater than or equal to S2? _____
6. Is DD less than 444? _____
7. Is X greater than Y? _____
8. Is A$ greater than or equal to 79? _____
9. Is P$ not equal to ''YES''? _____
10. Is VP less than or equal to JK? _____



PROGRAMMER'S PASTIME #44

The location of the IF–THEN statement in a program is very important. If it is put in the wrong place, the program won't work properly. The IF–THEN statement must come **after** the LET or INPUT statements that introduce the variables in the IF–THEN statement. For example:

Program

```
10 IF P<Q THEN 50
20 LET P=5
30 LET Q=7
40 GOTO 60
50 ?"P IS SMALLER"
60 END
```

Output

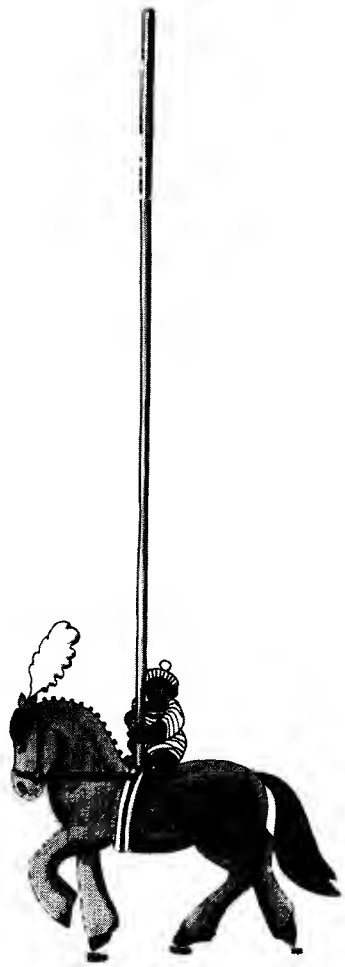
ATARI does not print anything because the IF–THEN statement is before the LET statements that introduce the variables.

In of the following programs, the IF–THEN statement is in the wrong place. Rewrite the programs so they are correct.

Incorrect program

1.
```
10 IF Z=2 THEN 60
20 LET A=6
30 LET B=8
40 LET Z=2
50 GOTO 70
60 ?"Z=2"
70 END
```

Corrected program



Incorrect program **Corrected program**

2.
```
 5 DIM E$(10),D$(10)
10 ? "WHAT COLOR
   ARE YOUR
   EYES?"
20 IF E$ = "BLUE"
   THEN 100
30 INPUT E$
40 ? "ARE THEY
   DIFFERENT
   COLORS?"
50 IF D$ = "YES"
   THEN 120
60 INPUT D$
70 ? "THEY ARE 1
   COLOR"
80 ? "THEY ARE NOT
   BLUE"
90 GOTO 130
100 ? "WHAT NICE
   BLUE EYES!"
110 GOTO 130
120 ? "WHAT
   COLORFUL
   EYES!"
130 END
```

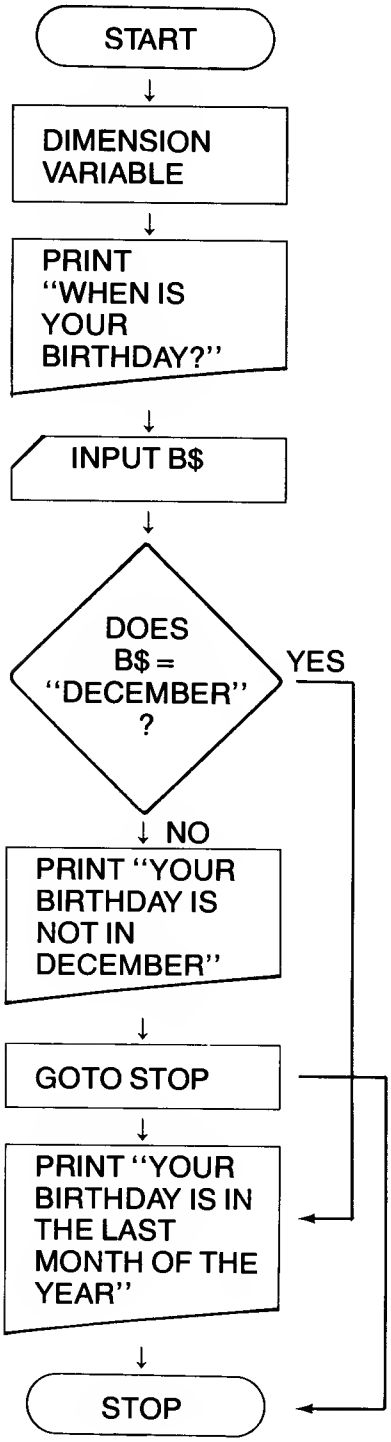
3.
```
 5 DIM F$(10)
10 IF F$ = "NO"
   THEN 60
20 ? "ARE
   COMPUTERS
   FUN?"
30 INPUT F$
40 ? "YOU'RE
   RIGHT!"
50 GOTO 70
60 ? "YOU'RE NO
   FUN!"
70 END
```

PROGRAMMER'S PASTIME #45

Study each flow chart, then write a program.
Debug your programs by running them on ATARI.

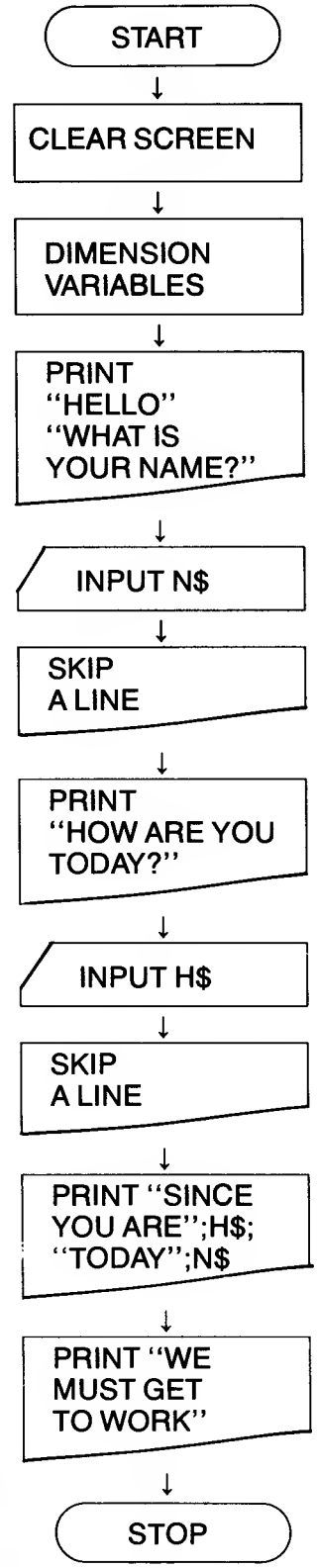
Flow Chart **Program**

1.



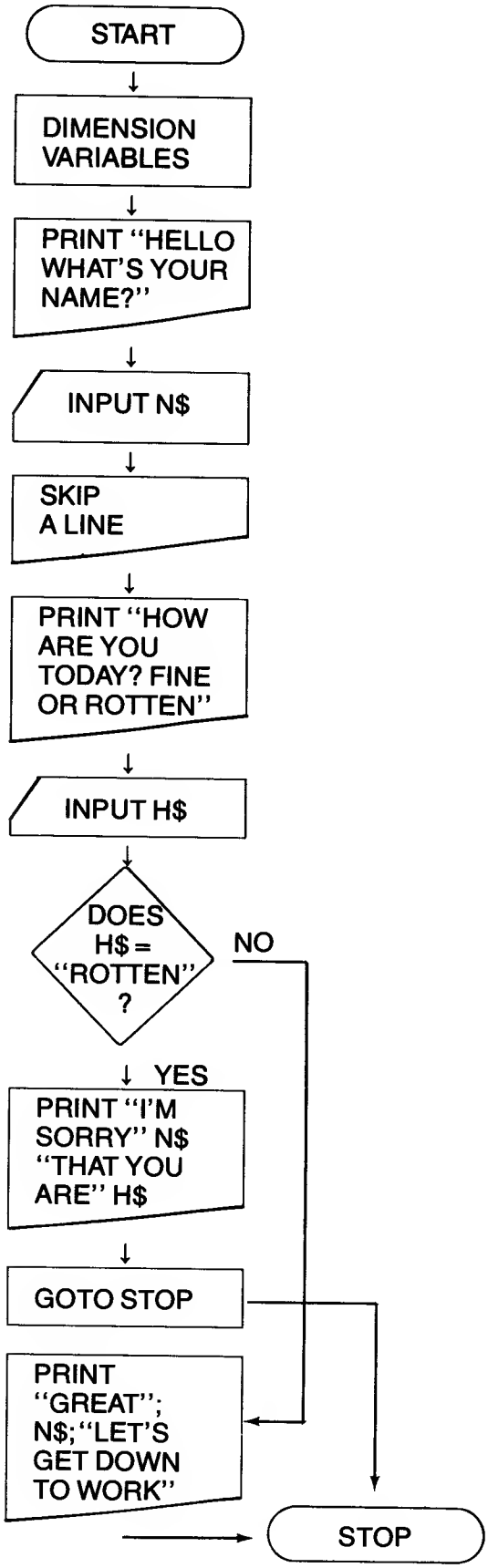
Flow Chart **Program**

2.



Flow Chart **Program**

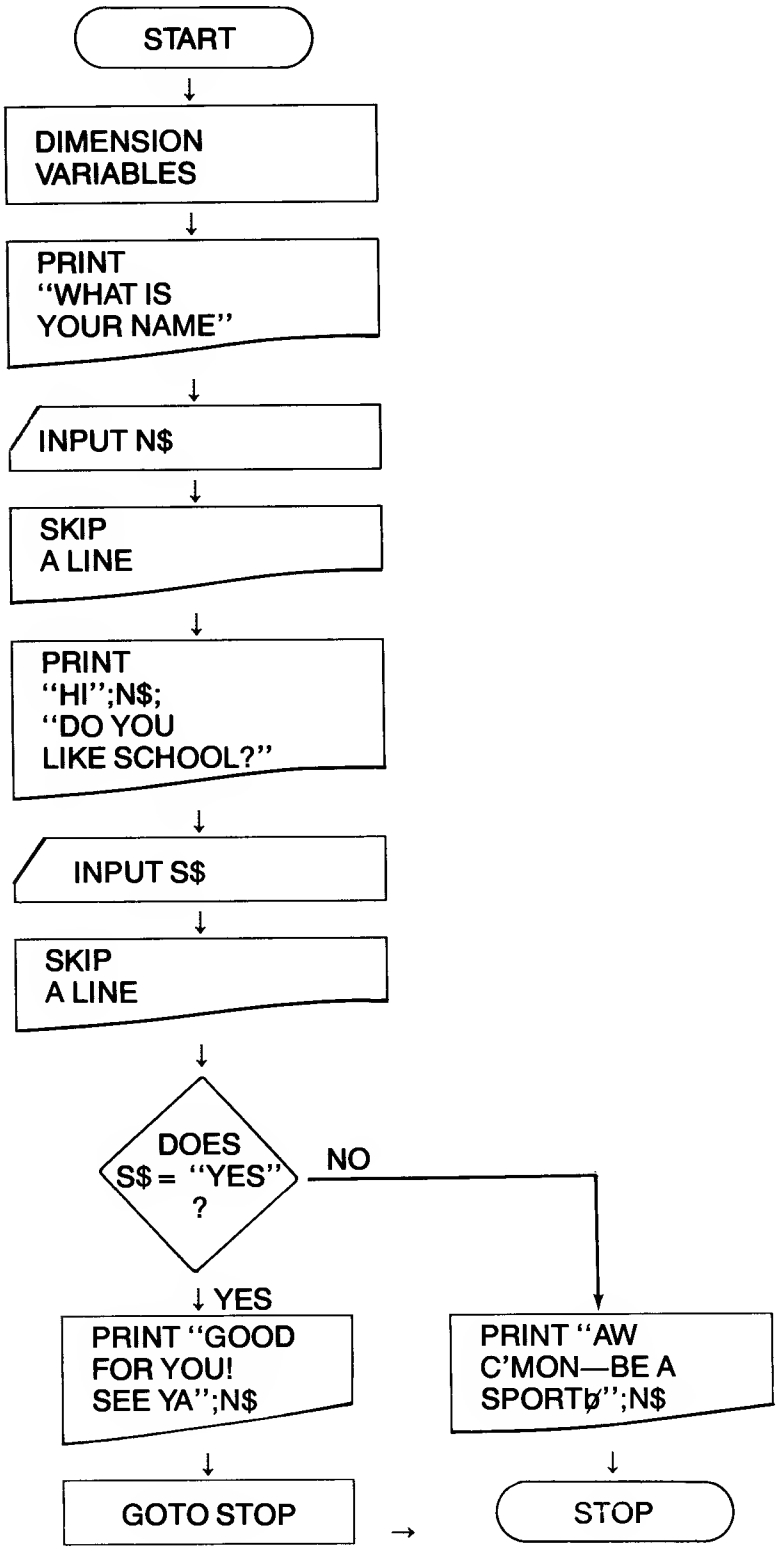
3.



Clue: You will need to use the complement of the question for your IF–THEN statement.

Flow Chart **Program**

4.



Clue: You will need to use the complement of the question for your IF–THEN statement.

PROGRAMMER'S PASTIME #46

For each description, write an algorithm in flow chart form and write a program for the flow chart. Debug each program by running it on ATARI.

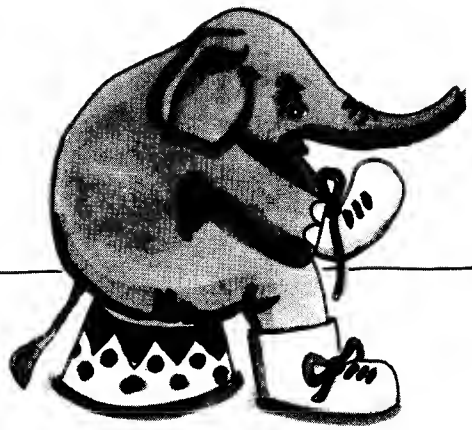
Description **Flow Chart** **Program**

1. Alphabetize "HIP" and "HIPPO"

2. Alphabetize "GUSTO" and "GROOVY"

Description **Flow chart** **PROGRAM**

3. Alphabetize
   "AARDVARK" and
   "ZEBRA"

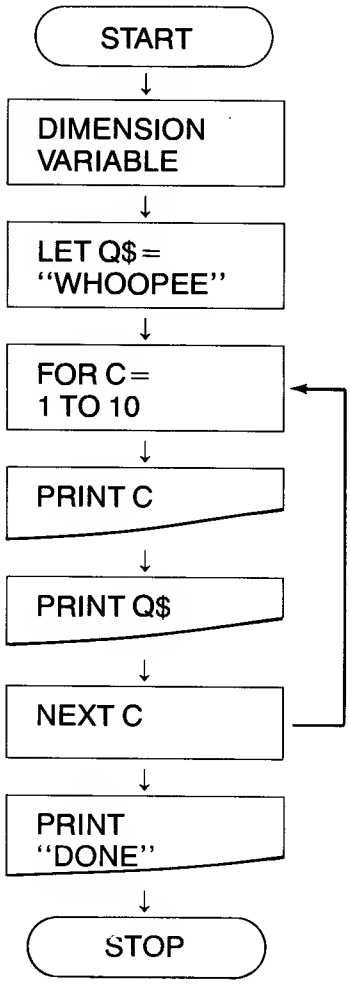


PROGRAMMER'S PASTIME #47

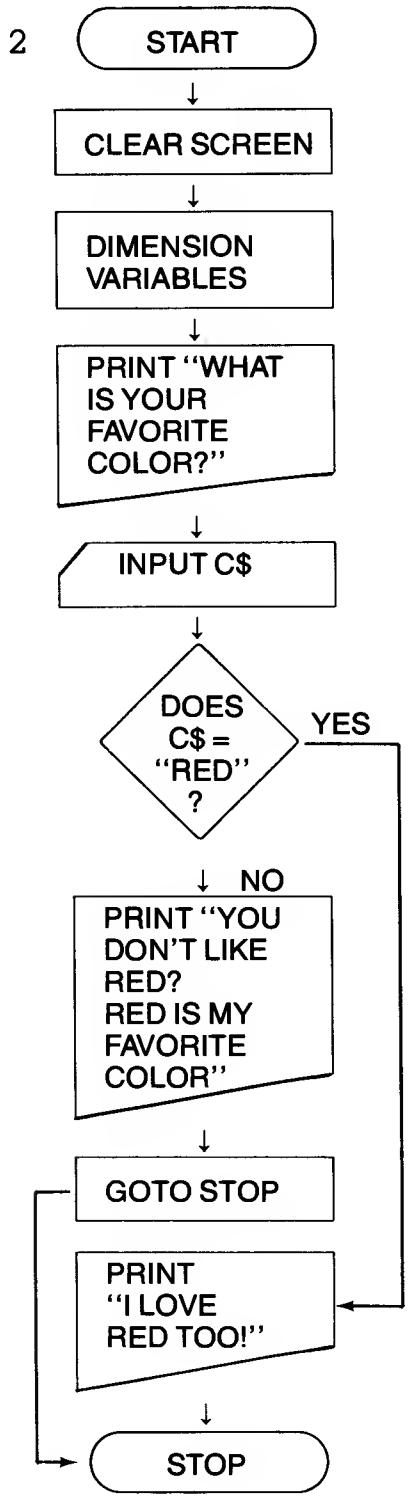
Study each flow chart and then write a program. Use REM statements where appropriate to show good programming style. Debug your programs by running them on ATARI.

Flow Chart

1.

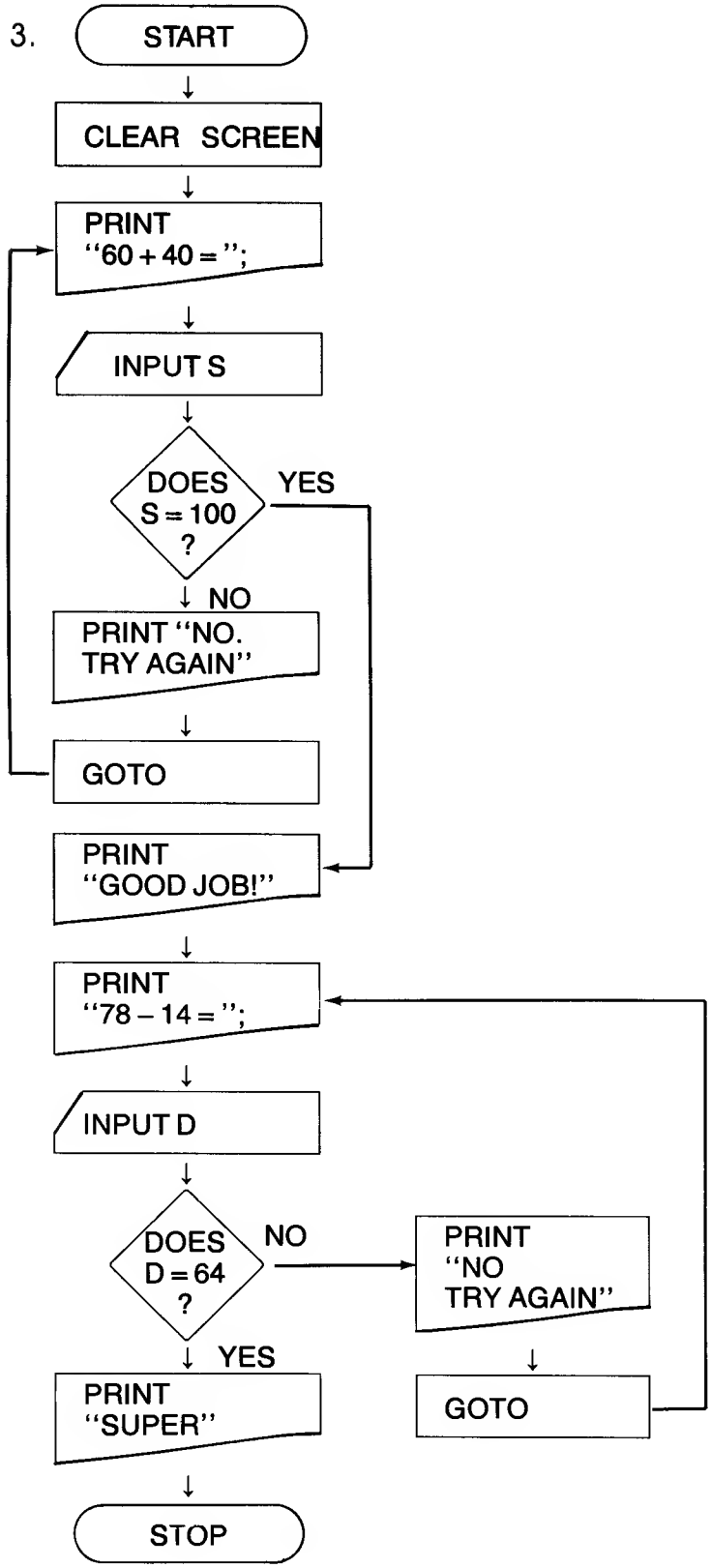


Flow Chart **Program**

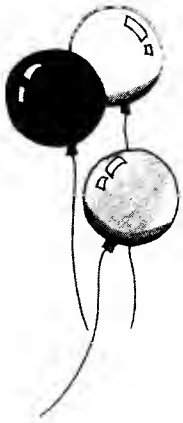


Flow chart **Program**

3.



PROGRAMMER'S PASTIME #48



Study each program and write what you think ATARI would print as the OUTPUT—including error messages. Check your answers by running the programs on ATARI.

Program **Output**

1.
```
 5 DIM Z$(10),L$(10)
10 READ Z$, L$
20 ? Z$, L$
30 GOTO 10
40 DATA "YOU",
   "ARE", "A",
   "HOT-SHOT"
50 END
```

2.
```
10 DATA 4,14,41,6,
   16,61,3,13,31
20 READ Q, R, S
30 ? Q+R+S
40 GOTO 20
50 END
```

3.
```
10 READ G, H
20 DATA 44,66,88,
   22,110
30 ? G, H
40 GOTO 10
50 END
```

4.
```
10 DATA 14,7,2,16,8,
   2,-99,-99,-99
20 READ A, L, B
30 IF A=-99 THEN
   60
40 ? A-L-B
50 GOTO 20
60 END
```

Program **Output**

5.
```
10 READ R1, R2
20 IF R1 = -1 THEN
     60
30 ?R$*R2
40 GOTO 10
50 DATA 2,2,3,3,4,
     4,5,5,-1,-1
60 END
```

6.
```
 5 DIM
D$(10),E$(10)
10 FOR L=1 TO 4
20 READ D$, E$
30 ?D$, E$
40 NEXT L
50 DATA "A", "E",
     "I", "O"
60 DATA "U", "Y",
     "ARE",
     "VOWELS"
70 END
```

7.
```
10 FOR L=1 TO 2
20 READ S1, S2, S3
30 DATA 8,2,4,6,2,3
40 ?S1*S2*S3
50 NEXT L
60 END
```

PROGRAMMER'S PASTIME #49

In each of the following programs there are mistakes. Circle the line number(s) with the mistake and make your correction in the space to the right. If something has been left out, add it to the program.

Program **Correction**

1.
```
10 READ P,A,N
20 ?P,A,N
30 DATA 400, 8%, 6
40 END
```

2.
```
 5 DIM N$(10)
10 READ N$, A
20 ?N$, A
30 DATA KIM IS,
   3*4
40 END
```

3.
```
 5 DIM N$(10)
10 ?"NAME",
   "AGE"
20 READ A, N$
30 ?A, N$
40 DATA HARVEY,
   14
50 END
```

4.
```
 5 DIM N$(10)
10 ?"NAME", "AGE"
20 READ N$, A
30 ?N$, A
40 DATA HARVEY,
   14 YEARS OLD
50 END
```

Program **Correction**

5.
```
 5 DIM F$(20)
10 READ F$
20 ?"DAILY MENU"
30 ?F$
40 END
```

6.
```
10 READ X,Y,Z
20 ?"THE PRODUCT
   OF 3 NUMBERS"
30 ?X*Y*Z
40 DATA 4,5
50 END
```

7.
```
10 ?"COUNTING"
20 READ DATA
30 DATA 1,2,3,4
40 END
```

8.
```
 5 DIM N$(10)
10 ?"NAME", "AGE"
20 READ N$, A
30 ?N$, A
40 GOTO 20
50 DATA BOB,
   BILL,10,11
```

PROGRAMMER'S PASTIME #50

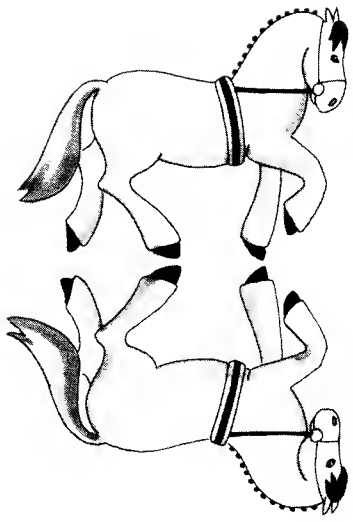
READ–DATA statements can help you write shorter programs. Rewrite each program using READ–DATA statements to shorten them. Try to write each program so you don't get an error message.

Long program

1.
```
10 ?"MULTIPLYING
   2 NUMBERS"
20 LET P=60
30 LET Q=129
40 LET R=410
50 LET S=.6
60 ?P,Q, P*Q
70 ?R,S, R*S
80 END
```

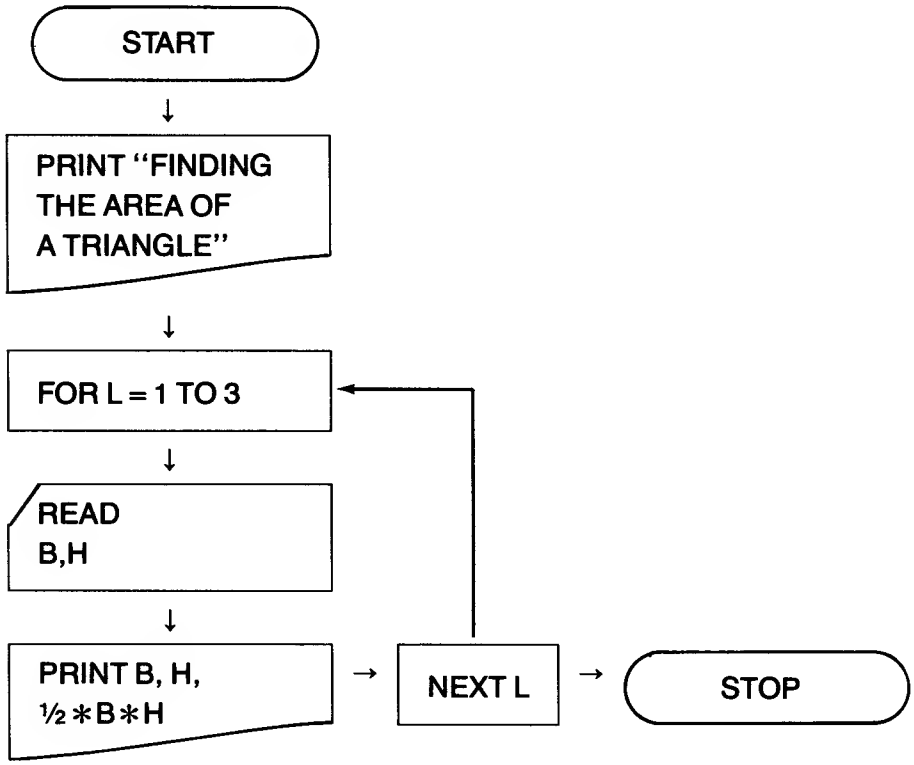
2.
```
 5 DIM A$(10),
   B$(10), C$(10),
   D$(10), E$(10)
10 ?"TEST SCORES"
20 ?"NAME",
   "SCORE"
30 LET A$= "JOE"
40 LET A=98
50 LET B$= "TOM"
60 LET B=52
70 LET C$= "KRIS"
80 LET C=95
90 LET D$= "GAIL"
100 LET D=75
110 LET E$= "BOB"
120 LET E=72
130 ?A$, A
140 ?B$, B
150 ?C$, C
160 ?D$, D
170 ?E$, E
180 END
```

Short program



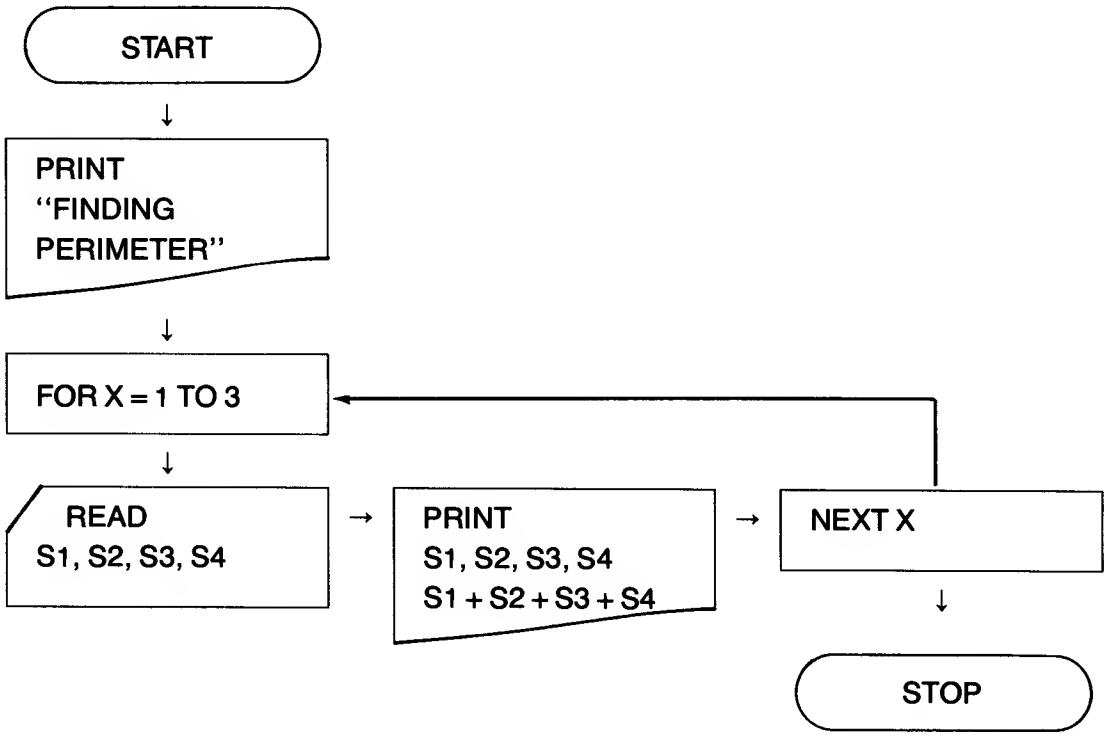
Flow Chart **Program**

4.

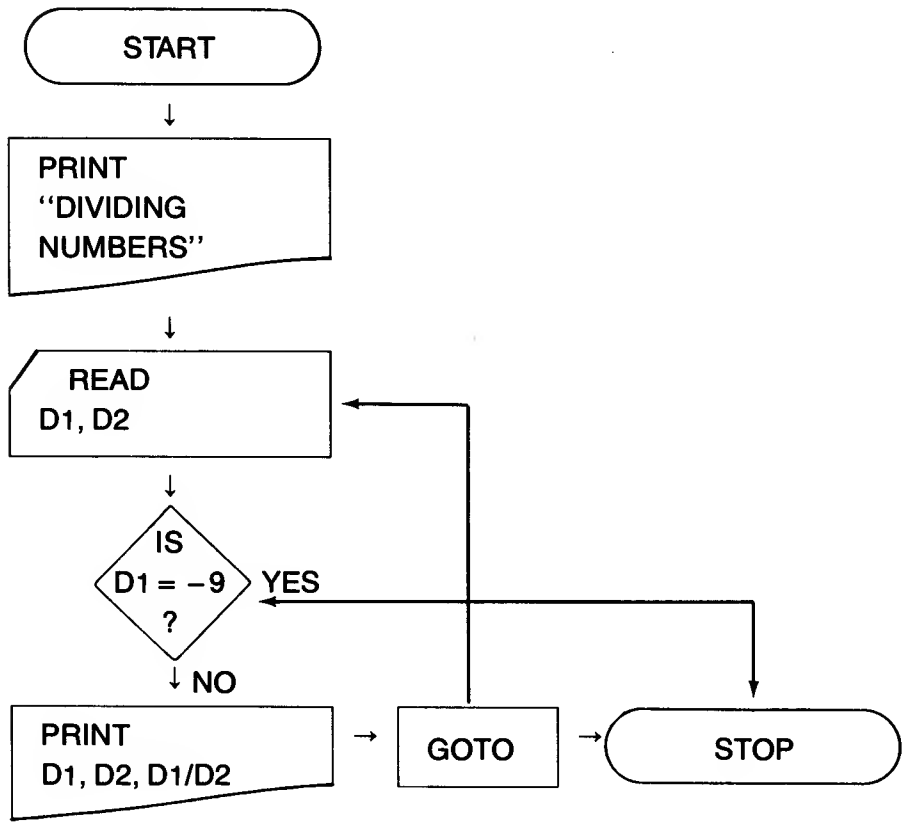


Flow chart **Program**

2.



3.



2. Write a program that lists the names of your friends.

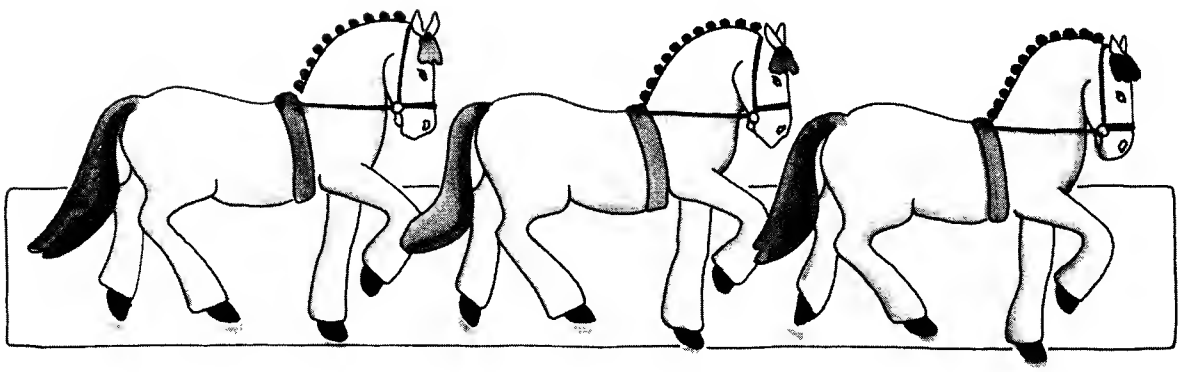
Flow chart **Program**

PROGRAMMER'S PASTIME #52

Using what you know about READ-DATA statements:

1. Write a program that multiplies three numbers.

Flow chart **Program**



4. Flow Chart

5. CODE the program

6. DEBUG
7. REVISE

PROGRAMMER'S PASTIME #53

Use the problem-solving approach to get ATARI to solve the following problems.

Problem 1

The teacher gave your class a test on programming the computer. The test scores were:

Jill Jarvis	73%	Your teacher needs to
Katie O'Keefe	98%	know the **average** test
Tommy Temple	67%	score.
Susie Sunbeam	82%	
You	90%	

Write a program that tells ATARI to calculate and print the average score.

HINT: To find the average of 5 numbers, add them together and divide by 5.

1. THINK about the problem.
2. Make your DATA TABLE here.

3. Write the ALGORITHM (steps and equations).

4. Flow Chart

5. CODE

6. DEBUG
7. REVISE

Problem 2

You are the new manager of the "Peppy Pizza" restaurant and you need the help of a computer. Write a program that will allow you to INPUT the number of small, medium, and large pizzas sold during a day.

Have ATARI print out the total number of pizzas sold and how much money you made.

PRICES:	small	$4.30
	medium	$5.50
	large	$7.25

OUTPUT HINT:

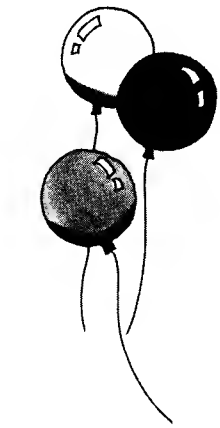
HOW MANY PIZZAS: (SMALL, MEDIUM, LARGE)
? _____, _____, _____
THERE WERE _____ PIZZAS SOLD TODAY.
"PEPPY PIZZA" MADE $_____.

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM

4. Flow Chart

5. CODE



6. DEBUG
7. REVISE

Problem 3

Write a program that will allow you to INPUT your age in years, months, and days. Example: 9 years, 3 months, 17 days.

Have ATARI calculate and print how many days, hours, and minutes old you are.

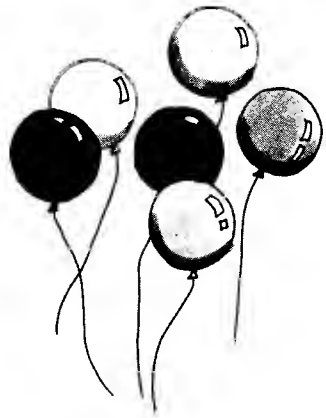
HINT: There are normally 365 days in a year and 30 days in a month. There are exactly 24 hours in a day and 60 minutes in an hour.

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM

4. Flow Chart

5. CODE



6. DEBUG
7. REVISE

Problem 4

You just got hired as a SUPER SCOOPER at the DIPPER DELIGHT Ice Cream Store. Write a program that will allow you to INPUT how many hours you worked for the week.

Have ATARI calculate and print hours worked and your salary for the week if you make $3.25 an hour.

OUTPUT HINT:

HOW MANY HOURS DID YOU WORK? _____

YOU WORKED _____ HOURS AND MADE $_____.

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM

4. Flow Chart

5. CODE

6. DEBUG
7. REVISE

Problem 5

Add to the problem you wrote for Problem 4 so that ATARI can calculate overtime pay. (Overtime is any hours worked **over** 40 hours a week.) You get paid $4.75 for every hour of overtime you work.

Add this to our OUTPUT:

YOU WORKED _____ OVERTIME HOURS AND
MADE $_____ IN OVERTIME.
YOUR TOTAL PAY FOR THE WEEK IS $_____.
(Total pay is regular pay + overtime pay.)

HINT: You will need a decision box in your flow chart to ask:

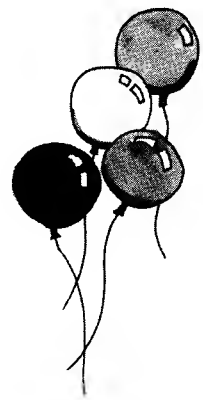
IS H > 40?

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM

4. Flow Chart

5. CODE



6. DEBUG
7. REVISE

Problem 6

You are the famous sportscaster H.E. Nosell. You have been asked to calculate the batting averages of Big League Baseball players. Write a program that allows you to INPUT a player's name, hits, and times at bat.

Have ATARI calculate and print the player's name and batting average.

HINT: To calculate batting average, use this equation:

1000*hits/times at bat

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM

Down

1. We can interact with the computer by using an _____ statement.
3. The statement used for making comparisons is _____-THEN.
5. The statements that let you change your data are the _____-DATA statements.
7. This sign, >, means _____ than.
9. A _____ table lists the variables you are using in a program.
11. An alphanumeric variable is also called a _____ variable.

Across

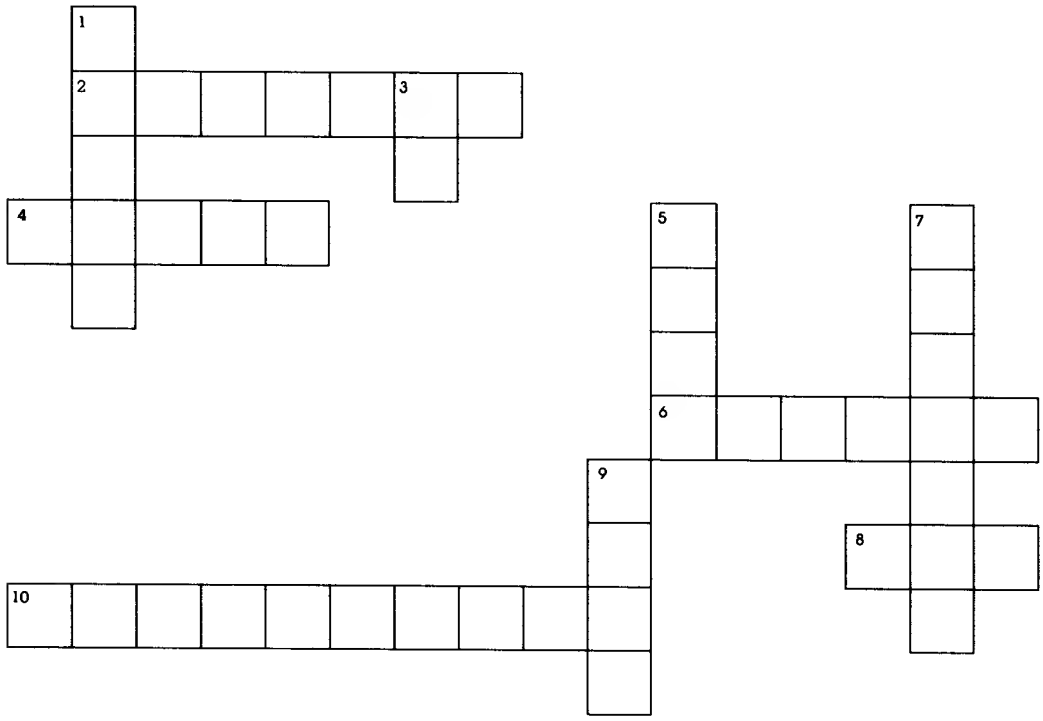
2. A variable that stores a number.
4. We use _____ data to let ATARI know that we are at the end of our data list.
6. A string variable is written as a letter followed by a _____ sign.
8. You should document your programs by using the _____ statement.
10. The opposite of a question is called its _____.
11. When ATARI is alphabetizing words, it knows that "A" is the _____ letter in the alphabet.

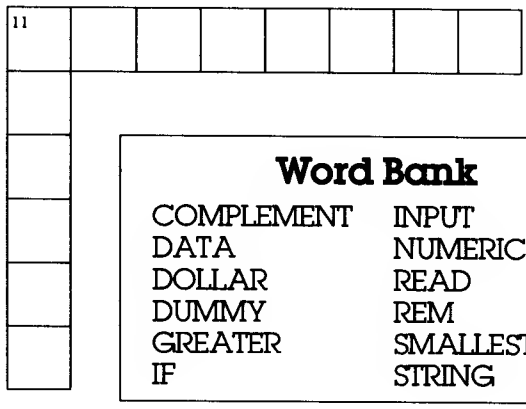
Evaluate Yourself

1. Component 6 was _____
   because _____
   _____
2. The best parts of the component were _____
   _____
   _____
3. The parts I liked the least were _____
   _____
   _____
4. The most valuable thing I learned in this component was _____
   _____
   because _____
   _____

Other comments:

COMPONENT 6 FUN PAGE





Word Bank

COMPLEMENT	INPUT
DATA	NUMERIC
DOLLAR	READ
DUMMY	REM
GREATER	SMALLEST
IF	STRING

Program **Important parts** **Output**

2.
```
10 REM CONVERT
   TEASPOONS TO
   TABLESPOONS
20 ?"TEASPOONS",
   "TABLESPOONS"
30 FOR T=3 TO 18
   STEP 3
40 ?T,,T/3
50 NEXT T
60 END
```

3.
```
10 REM CONVERT
   POUNDS TO OUNCES
20 ?"POUNDS",
   "OUNCES"
30 FOR P=1 TO 6
40 ?P,P*16
50 NEXT P
60 END
```

4.
```
10 REM CONVERT
   YARDS TO INCHES
20 ?"YARDS",
   "INCHES"
30 FOR Y=1 TO 5
40 ?Y,Y*36
50 NEXT Y
60 END
```

PROGRAMMER'S PASTIME #54

For each conversion problem, identify the important parts by writing: HEADING, FOR–NEXT LOOP, CONVERSION EQUATION next to the lines in the program. Then write what you think ATARI would print as the output.

Program Example:	**Important parts**	**Output**	
10 REM CONVERT FEET TO METERS		FEET	METERS
		1	0.3
20 ? "FEET", "METERS"	HEADING	2	0.6
30 ?		3	0.9
40 FOR F=1 TO 10	FOR–NEXT LOOP	4	1.2
50 ? F, F*.3	CONVERSION	5	1.5
60 NEXT F	EQUATION	6	1.8
70 END		7	2.1
		8	2.4
		9	2.7
		10	3

1.

10 REM CONVERT FEET TO YARDS	
20 ? "FEET", "YARDS"	HEADING
30 ?	
40 FOR F=1 TO 12 STEP 2	FOR–NEXT LOOP
50 ? F, F/3	CONVERSION
60 NEXT F	EQUATION
70 END	

Problem **Program**

4. Convert 1–10 liters to quarts.
   CONVERSION EQUATION:
   QUARTS=L/3.8

5. Convert 0°–100° Fahrenheit to Celsius.
   CONVERSION EQUATION:
   °C=5*(F–32)/9

6. Convert 1–100 pounds to kilograms.
   CONVERSION EQUATION:
   Kilograms=P*.45

PROGRAMMER'S PASTIME #55

Write a conversion program for each problem. Make sure your program has a heading, FOR-NEXT loop, and conversion equation. Run your programs on ATARI to check for bugs.

Problem **Program**

1. Convert 1-20 inches to centimeters.
   CONVERSION EQUATION:
   Centimeters=I*2.5

2. Convert 1-20 kilometers to miles.
   CONVERSION EQUATION:
   Miles=K/1.6

3. Convert 1-20 pounds to grams.
   CONVERSION EQUATION:
   Grams=P*454

4. Flow Chart

5. CODE

6. DEBUG
7. REVISE

PROGRAMMER'S PASTIME #56

Use the problem-solving approach to get ATARI to solve the following conversion problems.

A. Jed needs to find out what decimal $^{6}/_{7}$ stands for. Write a program that lists the fractions $^{1}/_{7}$ through $^{7}/_{7}$ and the decimals they stand for. CONVERSION EQUATION: Decimal = X/7

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM

4. Flow Chart

5. CODE

6. DEBUG
7. REVISE

B. Amy Astronaut is going to the moon. She learned that because the gravity on the moon is only 1/6 of the earth's gravity, she will weigh less on the moon. Write a program that asks you to INPUT how much you weigh. Then have ATARI print how much you would weigh on the moon. CONVERSION EQUATION: moon weight=earth weight/6

1. THINK about the problem
2. DATA TABLE

3. ALGORITHM

CHALLENGE

D. Fred's class took a test in which there were 20 questions asked. Fred's score was 16 correct out of 20, or $\frac{16}{20}$. Fred wants to know what percentage this would be. Write a program that lists the percentages for the test scores $\frac{1}{20}$ through $\frac{20}{20}$.

X = number answered correctly
20 = total number of questions

CONVERSION EQUATION: P=X*100/20

1. THINK about the problem
2. DATA TABLE

3. ALGORITHM

C. Add to program #2 so that ATARI will print a conversion table **after** printing the output for program #2. The table should list weight on earth from 10 to 100 pounds and the equivalent moon weights.

1. Flow Chart
2. CODE

3. DEBUG
4. REVISE

CHALLENGE

E. Change program #4 so ATARI asks you to INPUT how many test questions there were (T), and how many questions you answered correctly (C). Have ATARI print your score and the percentage you got correct.

HINT: score=C out of T
percentage=C*100 / T

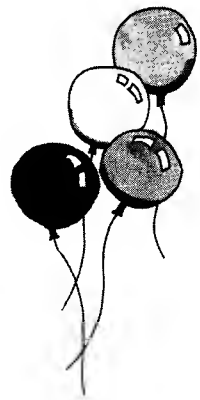
1. Flow Chart
2. CODE
3. DEBUG
4. REVISE

4. Flow Chart

5. CODE

6. DEBUG
7. REVISE

PROGRAMMER'S PASTIME #58



Read each RND function. Figure out what the lowest and highest random numbers will be that ATARI could print.

Function	**ATARI will print random numbers between and including:**
Example:	
LET X=18*RND(1)	0 and 17.9999
1. LET X=300*RND(1)	_____ and _____
2. LET X=RND(1)	_____ and _____
3. LET X=3*RND(1)	_____ and _____
4. LET X=67*RND(1)	_____ and _____
5. LET X=100*RND(1)+1	_____ and _____
6. LET X=25*RND(1)+1	_____ and _____
7. LET X=116*RND(1)+1	_____ and _____
8. LET X=39*RND(1)+1	_____ and _____
9. LET X=436*RND(1)	_____ and _____
10. LET X=77*RND(1)+1	_____ and _____
11. LET X=43*RND(1)+1	_____ and _____
12. LET X=13*RND(1)	_____ and _____
13. LET X=59*RND(1)+1	_____ and _____

PROGRAMMER'S PASTIME #57

RUN the program three times on ATARI. Each time the program is run, write down the random numbers that ATARI printed. Then write the lowest and highest numbers in the list. Run the program several more times and visually note the highest and lowest numbers.

RUN #1

numbers					
lowest					
highest					

Program

```
10 FOR L=1 TO 5
20 LET X=10*RND(1)
30 ?X
40 NEXT L
50 END
```

RUN #2

numbers					
lowest					
highest					

RUN #3

numbers					
lowest					
highest					

PROGRAMMER'S PASTIME #60

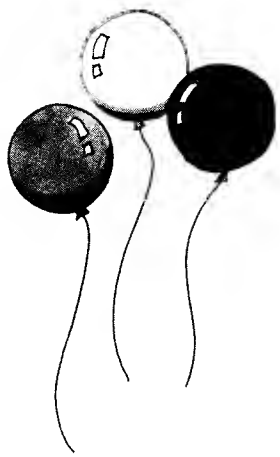
We use both the INT and RND functions to tell ATARI to print a random integer. Read each function. Then write the two numbers that ATARI must create random integers **between**.

Function	**ATARI will print random integers between:**
Example:	
INT(40*RND(1)+26)	26 and 67

DO: 40+26+1=67

1. INT(14*RND(1)+3) _____ and _____
2. INT(221*RND(1)+99) _____ and _____
3. INT(3*RND(1)+2) _____ and _____
4. INT(22*RND(1)+16) _____ and _____
5. INT(55*RND(1)+28) _____ and _____
6. INT(77*RND(1)+75) _____ and _____
7. INT(94*RND(1)+33) _____ and _____
8. INT(101*RND(1)+66) _____ and _____
9. INT(63*RND(1)+7) _____ and _____
10. INT(80*RND(1)+45) _____ and _____
11. INT(46*RND(1)+23) _____ and _____
12. INT(39*RND(1)+19) _____ and _____

PROGRAMMER'S PASTIME #59



INTEGERS are whole numbers. The INT function rounds DOWN to the next whole number to make it an integer. Read each INT function. Then write what ATARI would print for the output.

Function	**Output**
Example: 10 LET C=4.96 20 ?INT(C)	4
1. 10 LET X=66.823 20 ?INT(X)	
2. ?INT(4.89)	
3. 10 LET R=992.01 20 ?INT(R)	
4. ?INT(63.49321)	
5. 10 LET BD=−16.003 20 ?INT(BD)	
6. 10 LET P1=43.001 20 ?INT(P1)	
7. ?INT(660.666)	
8. ?INT(−33.23)	
9. 10 LET S=4120.7 20 ?INT(S)	
10. ?INT(−999.999)	

PROGRAMMER'S PASTIME #62

Write an INT and RND function for each description. Remember the equation:

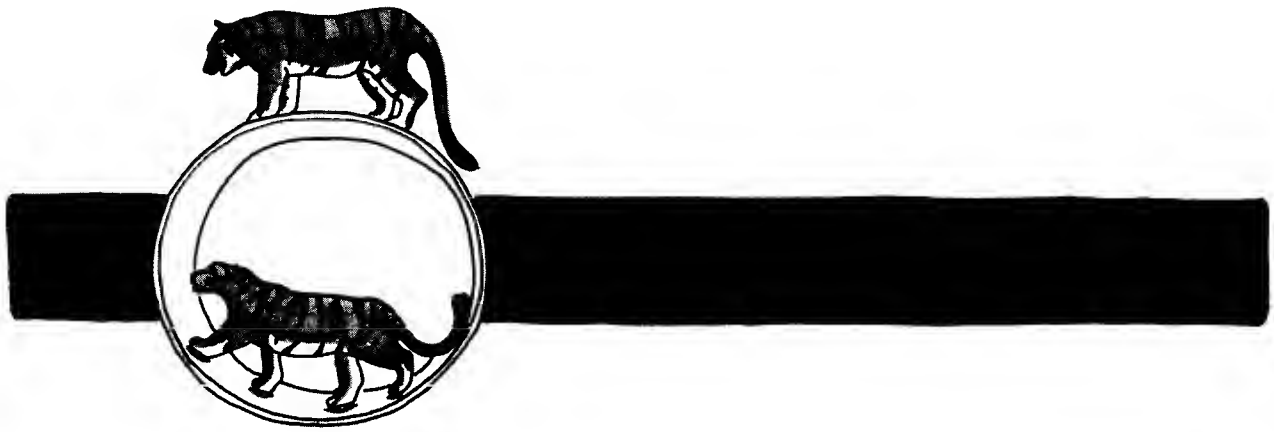
INT ((B−(A+1))*RND(1)+A)

B=largest number A=smallest number

To print random integers between	**Function**
Example: 5 and 18	INT(12*RND(1)+5)

DO: INT((18−(5+1))*RND(1)+5)
INT(12*RND(1)+5)

1. 16 and 48 _____
2. 2 and 10 _____
3. 10 and 100 _____
4. 1 and 50 _____
5. 33 and 99 _____
6. 50 and 100 _____
7. 75 and 100 _____
8. 27 and 41 _____
9. 62 and 300 _____
10. 49 and 52 _____



PROGRAMMER'S PASTIME #61

Write an RND function for each description.

Create random numbers between and including:	**Function**
Example:	
0 and 9.9999	LET X= 10*RND(1)
1. 0 and 14.9999	_____
2. 0 and 92.9999	_____
3. 1 and 45.9999	_____
4. 0 and 70.9999	_____
5. 1 and 26.9999	_____
6. 0 and 106.9999	_____
7. 0 and 66.9999	_____
8. 1 and 211.9999	_____
9. 1 and 31.9999	_____
10. 1 and 441.9999	_____
11. 0 and 89.9999	_____
12. 1 and 53.9999	_____
13. 1 and 382.9999	_____
14. 0 and 554.9999	_____

2. Write a CAI program that asks a student to multiply two random numbers between 1 and 10.

Flow chart **Program**

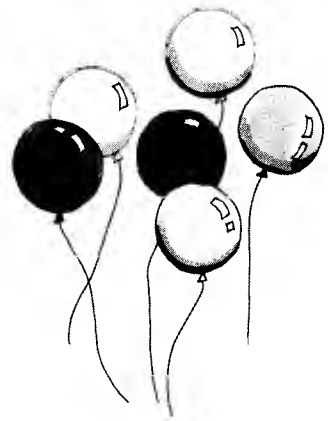
PROGRAMMER'S PASTIME #63

Make a flow chart and write a program for each problem. Debug your programs by running them on ATARI.

1. Write a program that will print 10 random decimals between 1 and 100 and then print the integer for each.

Flow Chart **Program**

PROGRAMMER'S PASTIME #64



1. Write your own music by adding the necessary DATA in line 60:

```
10 REM WRITE A SONG
20 READ N
30 SOUND 1, N, 10, 8
40 FOR T=1 TO 150:NEXT T
50 GOTO 20
60 DATA
```

2. The above program holds the last note until you press [BREAK], type END and press [RETURN]. Add to the above program so that it ends by itself. (HINT—use some dummy data in line 60.)

3. You can use the INPUT statement in a program with the RND and INT functions. Write a program so that ATARI will ask you to type in two integers. Then have ATARI print 10 random integers between those two numbers.

OUTPUT HINT:

```
TYPE IN TWO NUMBERS
AND I WILL CREATE TEN
RANDOM INTEGERS BETWEEN
THOSE TWO NUMBERS
?
10 RANDOM INTEGERS
   BETWEEN      AND      ARE:
```

Flow chart **Program**

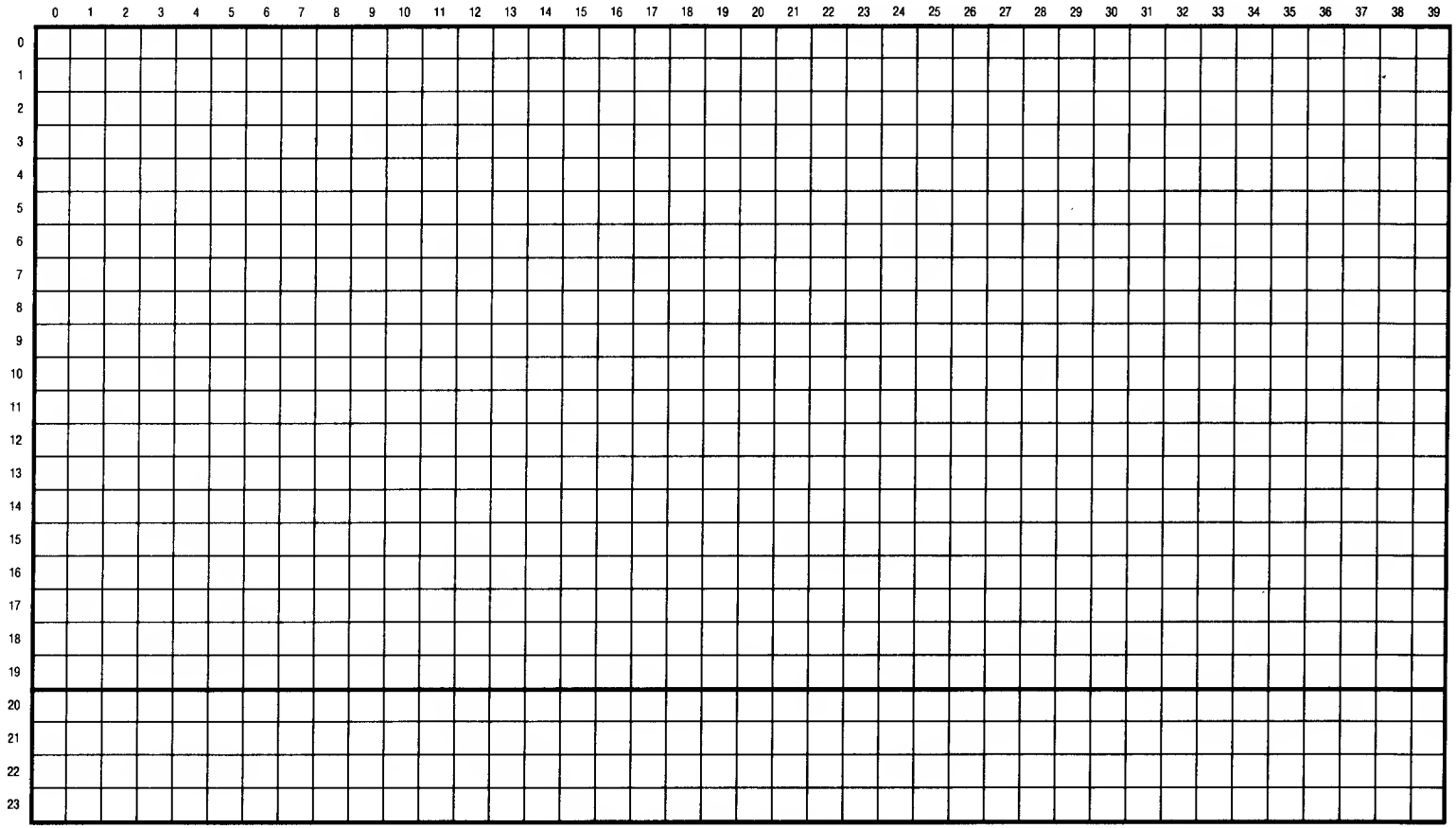
PROGRAMMER'S PASTIME #65

1. Select a simple song from a music book (Mary Had a Little Lamb, Jingle Bells, etc.), and write a program so that the song can be played by ATARI. (HINT—Use this formula and set in the proper Notes, SOUND: 0, N, 10, 8. Later, vary the Voice, Tone, and Loudness to see what happens.)

2. Write a program so ATARI will play music that you have composed.

3. Rewrite the program so that Voice, then Tone, and then Loudness are altered. (Be careful when changing Loudness so that others around you are not disturbed.)

4. Rewrite the program so that ATARI plays Notes randomly. (HINT—drop the READ/DATA statements and use a LET statement to assign a random number to N.)



GRAPHICS 3 SCREEN
(20×40—WITH TEXT WINDOW)

PROGRAMMER'S PASTIME #66

One of the most important aspects of producing good graphics, is being able to place the points and lines exactly where you want them. The key to doing so is to exactly locate a point by its column (X Coordinate) and row (Y Coordinate) position.

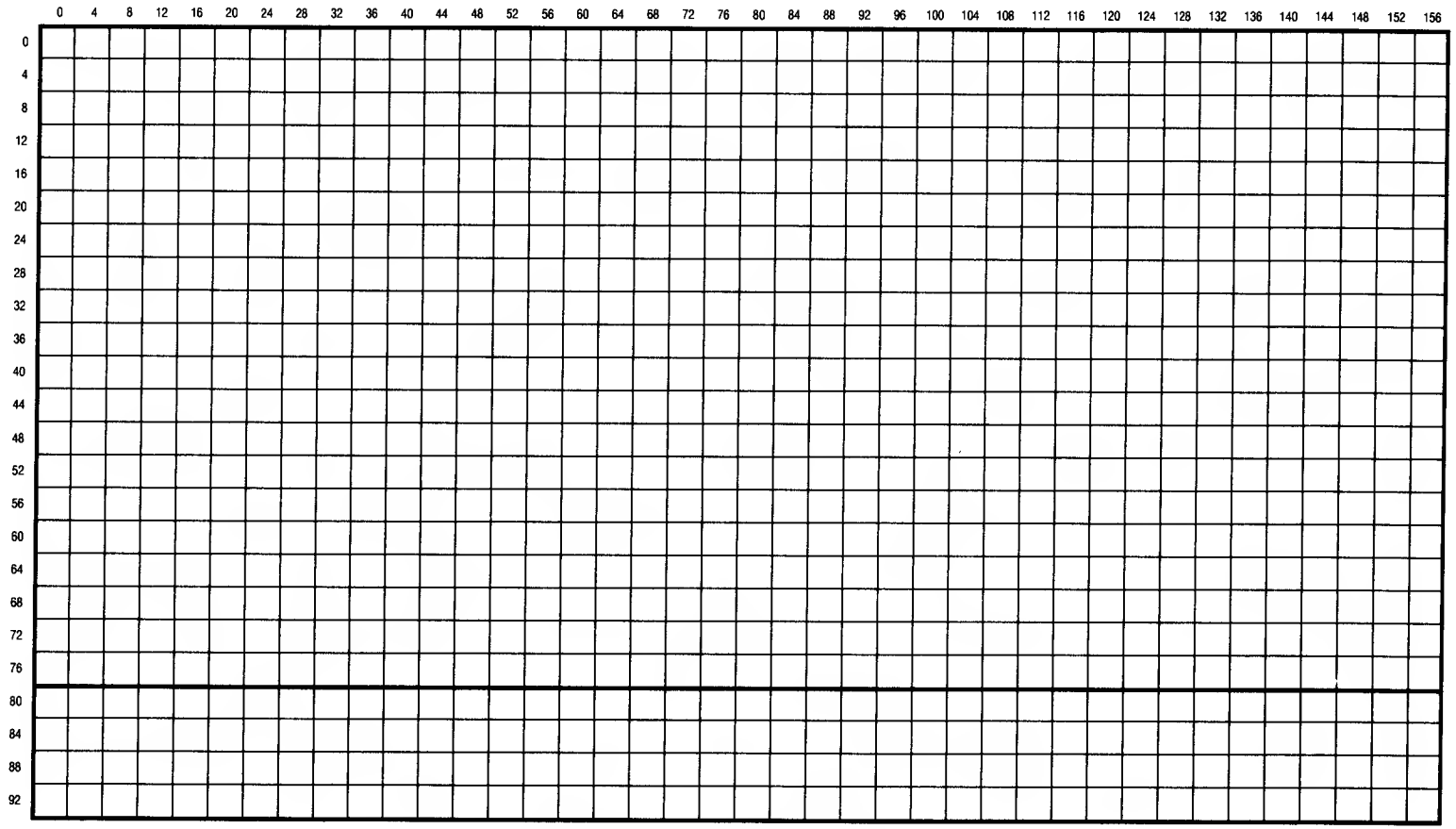
Use graph paper or the graphic screen illustrations that are on the next pages. Locate the following points on the Graphics 3 and Graphics 6 & 7 screens. Check your answers on ATARI. (Note — Although the two worksheet screens look similar, the numbering systems are different.)

Graphics 3

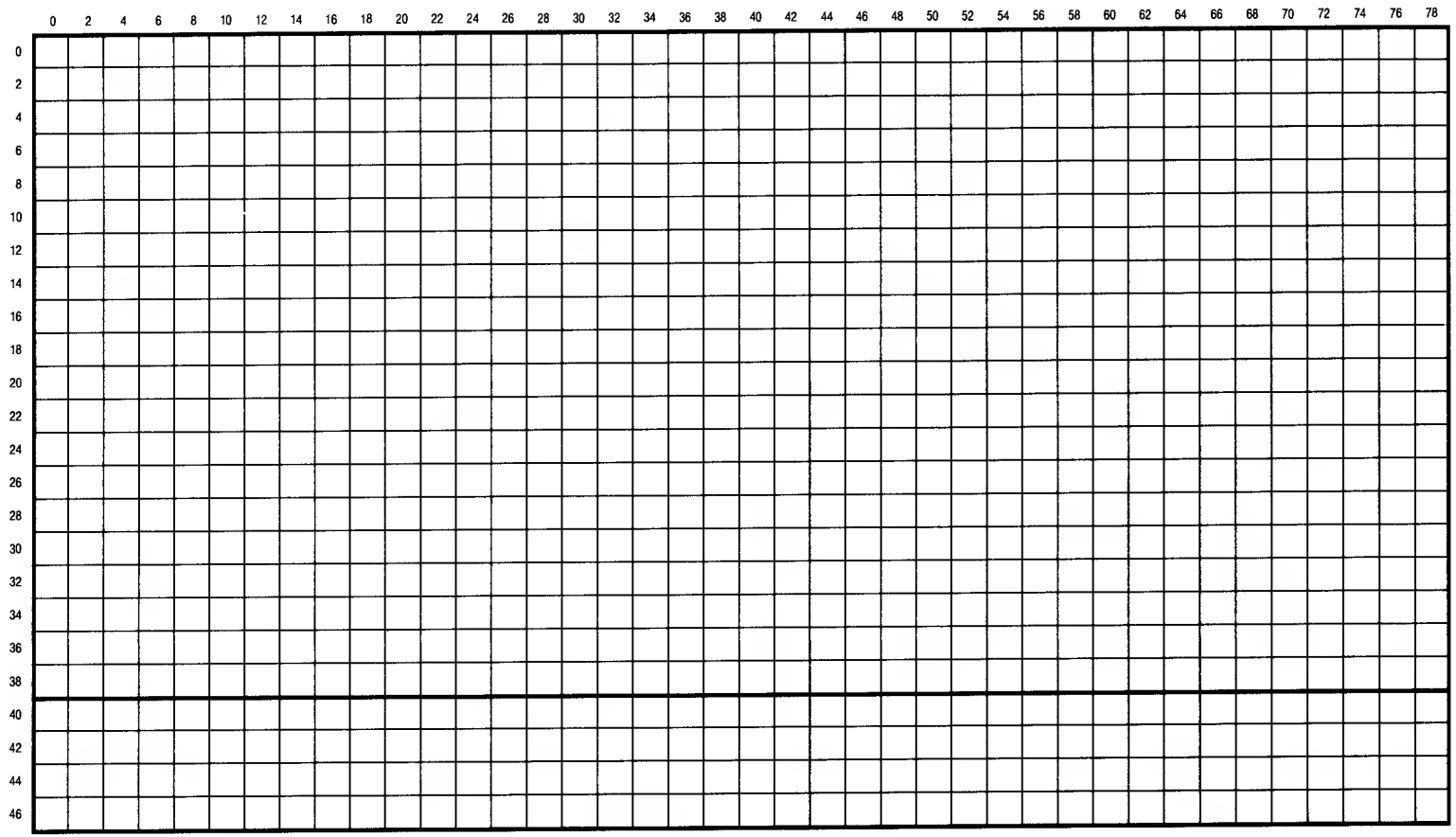
1, 10 (shown)
10, 1 (shown)
0, 0
5, 19
0, 19
19, 16
39, 0
39, 19
15, 22
10, 24

Graphics 7

1, 10 (shown)
10, 1 (shown)
0, 0
5, 19
0, 19
0, 79
159, 0
159, 79
60, 40
76, 161
76, 197



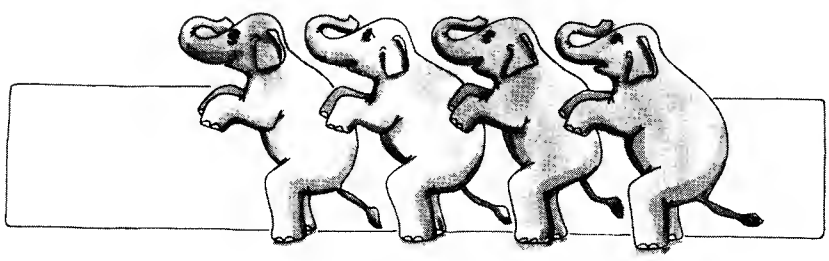
GRAPHICS 6 & 7 SCREEN
(80×160—WITH TEXT WINDOW)



GRAPHICS 4 & 5 SCREEN
(40×80—WITH TEXT WINDOW)

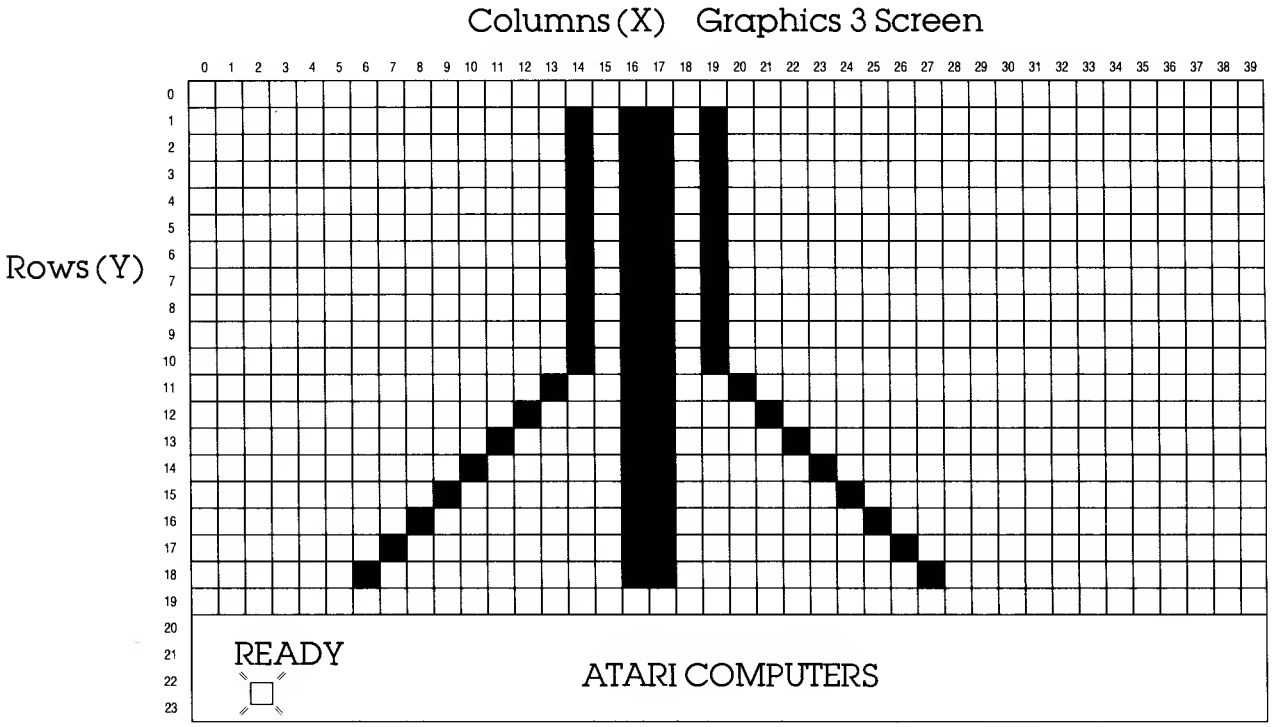
1. Try modifying the above program by adding COLOR statements at various lines, and by changing some of the DRAWTO and PLOT statements.

2. GET several pieces of graph paper from your teacher, or a reuseable piece that has been laminated, and draw some pictures. Convert the drawings to programs, and try them on ATARI. (It's often easiest to begin with simple drawings for the Graphics 3 screen. Just be sure that you use the appropriate graph paper for the Graphics Mode you want to use.)



PROGRAMMER'S PASTIME #67

One of the most enjoyable aspects of graphics is the ability to draw pictures. The key to doing so is to lay out a drawing on graph paper, and then convert the graph dimensions to statements that ATARI can understand. Notice the following drawing.



Here's how the drawing can be programmed for ATARI to understand:

```
10 GRAPHICS 3
20 COLOR 3
30 PLOT 14,1
40 DRAWTO 14,10
50 DRAWTO 6,18
60 PLOT 16,1
70 DRAWTO 16,18
80 PLOT 17,1
90 DRAWTO 17,18
100 PLOT 19,1
110 DRAWTO 19,10
120 DRAWTO 27,18
130 ?"          ATARI COMPUTERS"
```

PROGRAMMER'S PASTIME #69

1. It takes a lot of practice to know all the graphic variations you can create with ATARI. Using the following program, experiment by changing the COLOR, GRAPHICS MODE, and SETCOLOR factors.

```
10 GRAPHICS 3
20 COLOR 1
30 FOR X=0 TO 15
40 SETCOLOR 0,X,2
50 PLOT 5,5
60 DRAWTO 25,5
70 FOR T=1 TO 600:NEXT T
80 ?:?:?X
90 NEXT X
100 END
```

2. Take some graphic programs you have already written, or make some new ones, and improve them by using the SETCOLOR statement.

PROGRAMMER'S PASTIME #68

One enjoyable aspect of graphics is animation —causing the graphics to move. Following is a program for some simple animation.

```
10 GRAPHICS 3
20 COLOR 1
30 FOR X=0 TO 39 STEP 3
35 ?#6;" ESC SHIFT CLEAR < "
40 PLOT X,7
50 DRAWTO X,10
60 DRAWTO X+3,10
70 DRAWTO X+3,7
80 DRAWTO X,7
85 FOR T=1 TO 100:NEXT T
87 IF X>=36 THEN GOTO 10
90 NEXT X
```

Here's what the program does. Lines 40 through 80 make a simple square graphic. The X Coordinate is not specified in these lines, but rather it is set as a variable X. Lines 30 and 90 make the X Coordinate as every third number between 0 and 39. These lines, along with 10 and 20, which determine the graphics mode and color, are the main part of this program. However, notice how the program was improved by adding some more lines after the program was first written. Line 85 is a "timer" so that the graphic remains momentarily on the screen. (Try changing this line for different effects.) Line 87 causes the program to repeat once the graphic has moved completely across the screen. Line 35 causes the screen to be cleared as the graphic starts over. (Remember, #6 must be used with a PRINT statement for the graphics screen!)

1. Run this program, and then modify some line statements to see how you can change the graphics and animation.
2. Use all the graphic techniques that you have learned to this point to make your own animated graphics.

PROGRAMMER'S PASTIME #71

Use what you know about a good game program to write the game programs described below.

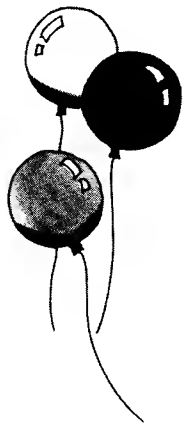
1. It is more meaningful to the user when the computer calls him or her by name—it makes the interaction more personal.

Write a GUESS A NUMBER game program that asks the user's name and calls the user by name throughout the program.

1. THINK about the program
2. DATA TABLE

3. ALGORITHM

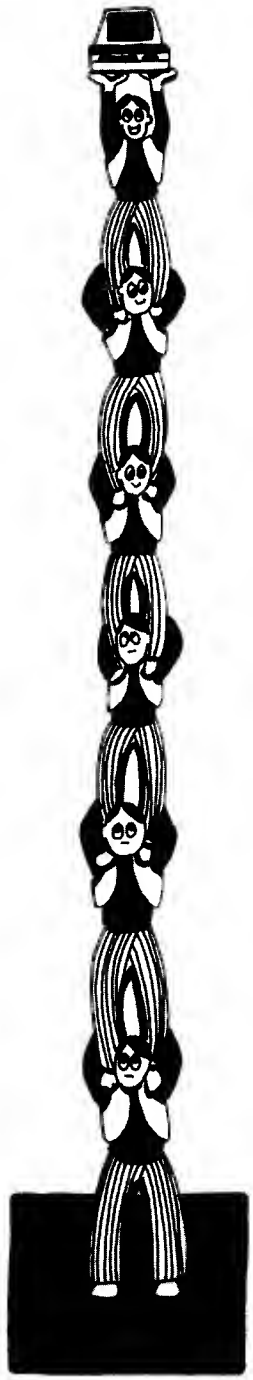
PROGRAMMER'S PASTIME #70



Using all the techniques you have learned for graphics, sound, and regular programming, create a fantastic light and sound show!

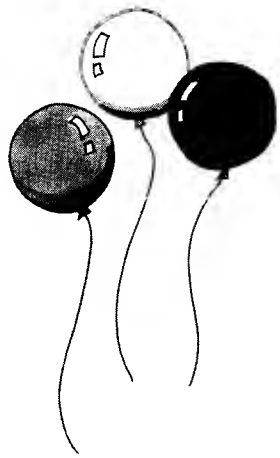
2. Revise the game program in #1 so a player must answer "YES" or "NO" in line 150.

If anything else is typed for INPUT, make ATARI print the question in LINE 140 again. This helps make the program GOOF PROOF.



4. Flow Chart

5. CODE



6. DEBUG
7. REVISE

4. Add something to the game program in #3 so ATARI asks the user the top number in the range they wish to guess. (For example, 1 to ____?)

After the user types in the top number, use it in the RND function to create a random integer between 1 and the top number.

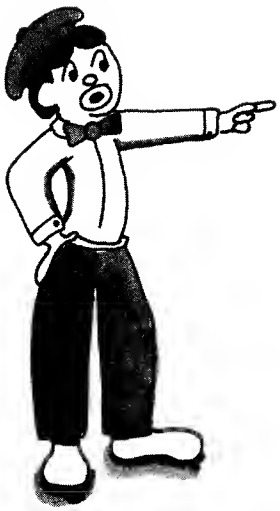
Hint: IF N=the top number
THEN you would use this RND function:
LET X= INT(N*RND(1)+1)

3. Add something to the game program in #2 so ATARI tells the user how many tries it took before they guessed the correct number.

HINT: Use a COUNTER, C.

Set C at 0 before the first guess. After the first guess add 1 to the counter: C=C+1.

Then after each of the next guesses, make sure one more is added to the counter. When the correct number is guessed, make ATARI print IT TOOK YOU _____ GUESSES.



4. Flow Chart

5. CODE

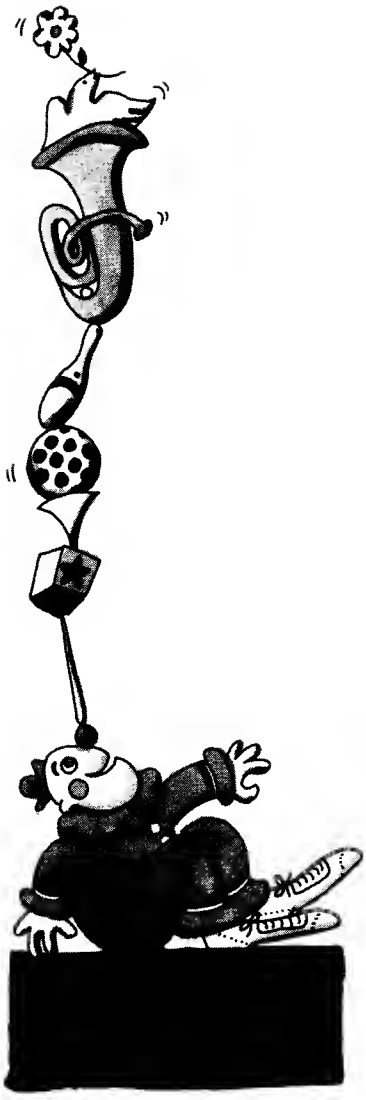
6. DEBUG
7. REVISE

5. Make up a computer game that uses a die. Write a program using all of the good style techniques you've learned.

The RND function for the throw of your die must choose a random integer between 1 and 6.

Example

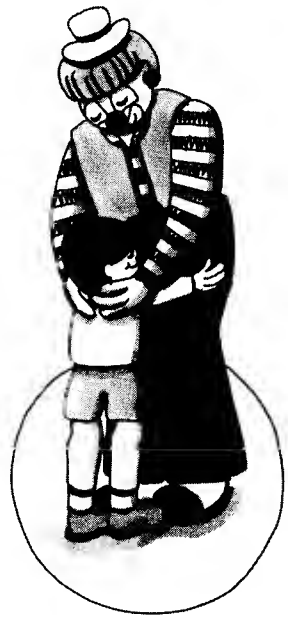
1. THINK about the program.
2. DATA TABLE



3. ALGORITHM

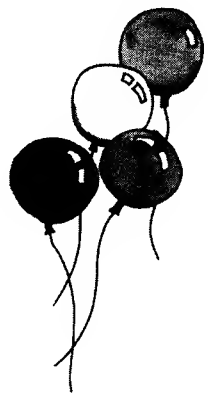
4. Flow Chart

5. CODE



6. DEBUG
7. REVISE

6. Create a computer program for any game you like. Include the five things every good game program should have. Try using animation. Be creative!



1. THINK about the program
2. DATA TABLE

3. ALGORITHM

Down

1. X and Y positions are called _____.
3. Write your program to be user-_____.
5. A word meaning "having no pattern or specific purpose:" _____.
7. C. stands for the _____ statement.
9. "Convert" means to _____.
11. The function which creates whole numbers in a program: _____.
13. This stands for a random function: _____

Across

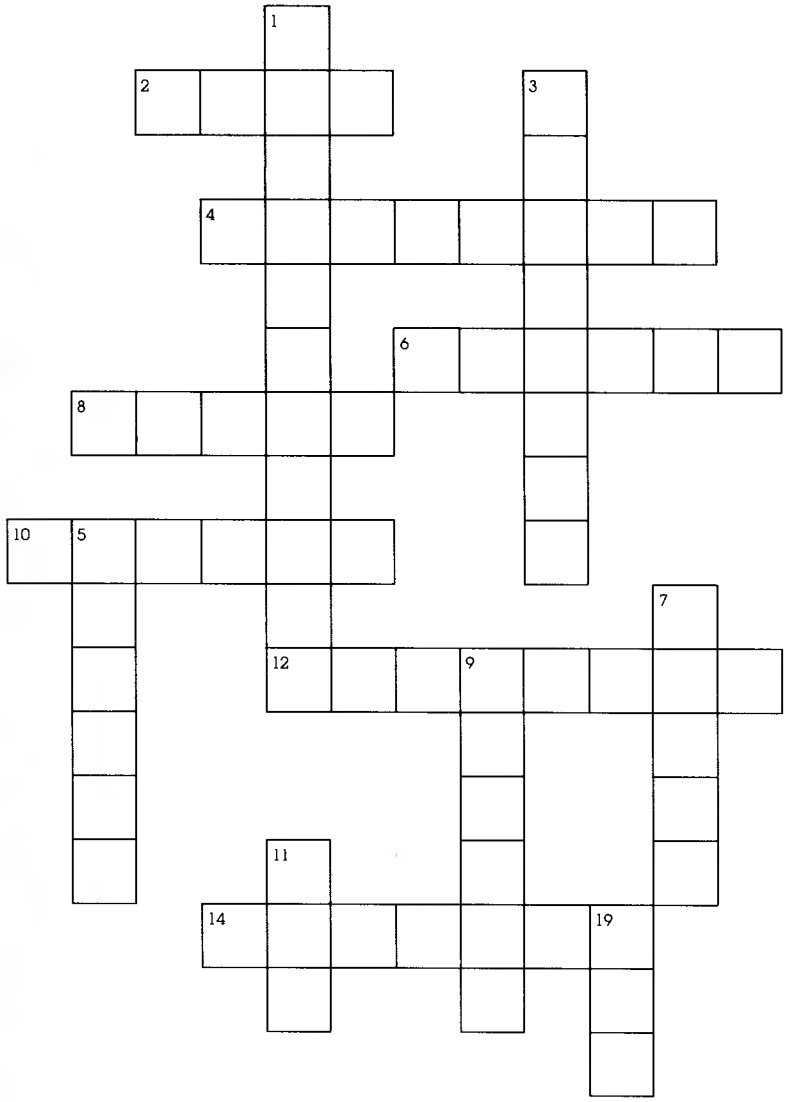
2. The statement which tells ATARI to place a point on the screen at a certain location: _____
4. GR. stands for the _____ statement.
6. The text _____ makes up the lower four lines of the screen in Graphics Mode.
8. We use this command when we want to hear ATARI: _____
10. The statement which tells ATARI to connect two points is _____.
12. If we want to change the color of our graphics or screen we must use the _____ statement.
14. Another name for a whole number is _____.

Evaluate Yourself

1. Component 7 was _____
because _____
_____
2. The best parts of the component were _____
_____
_____
3. The parts I liked the least were _____
_____
_____
4. The most valuable thing I learned in this component was _____
because _____
_____

Other comments:

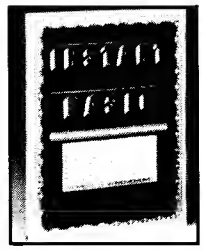
COMPONENT 7 FUN PAGE



Word Bank

CHANGE	INTEGER
COLOR	PLOT
COORDINATES	RANDOM
DRAWTO	RND
FRIENDLY	SETCOLOR
GRAPHICS	SOUND
INT	WINDOW

MORE HELPFUL WORDS FOR YOU from dilithium Press

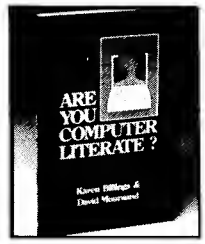


Instant (Freeze-Dried Computer Programming in) BASIC—2nd Astounding! Edition

Jerald R. Brown

Here is an active, easy, painless way to learn BASIC. This primer and workbook gives you a fast, working familiarity with the real basics of BASIC. It is one of the smoothest and best-tested instructional sequences going!

ISBN 0-918398-57-6 *200 pages* *$12.95*

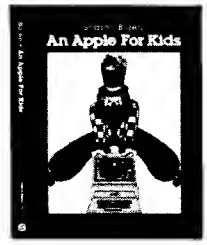


Are You Computer Literate?

Karen Billings and David Moursund

This is a fun introduction to computers for the real novice. Written by educators, this is a book that teaches the capabilities, limitations, applications and implications of computers.

ISBN 0-918398-29-0 *160 pages* *$9.95*

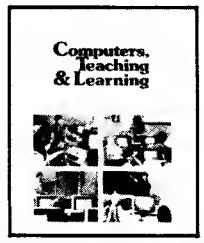


An Apple® For Kids / A PET® For Kids

Sharon Boren

A fresh, instructive, fun approach to teaching kids programming and computer operation. Written by a teacher, these books teach good programming skills with a focus on problem solving, improved thinking skills and creativity.

A PET® For Kids: *ISBN 0-88056-106-8* *148 pages* *$7.95*
An Apple® For Kids: *ISBN 0-88056-119-X* *148 pages* *$7.95*
Classroom Activity Workbook and Teachers Manual also available.



Computers, Teaching and Learning

Jerry Willis, D. Lamont Johnson and Paul Dixon

For the educator interested in using computers or the computer-user interested in educational applications, this book covers most of the important topics of concern to the educational environment.

ISBN 0-88056-065-7 *200 pages* *$9.95*

BRAINFOOD — Our catalog listing of over 130 microcomputer books covering software, hardware, business applications, general computer literacy and programming languages.

dilithium Press books are available at your local bookstore or computer store. If there is not a bookstore or computer store in your area, charge your order on VISA or MC by calling our toll-free number, (800) 547-1842.

Send to: dilithium Press, P.O. Box E, Beaverton, OR 97075

Please send me the book(s) I have checked. I understand that if I'm not fully satisfied I can return the book(s) within 10 days for full and prompt refund.

___ Instant (Freeze-Dried Computer Programming in) BASIC—2nd Astounding! Edition
___ Are You Computer Literate?
___ A Pet® For Kids
___ An Apple® For Kids
___ Computers, Teaching and Learning

☐ Check enclosed $ _____
Payable to dilithium Press

☐ Please charge my
VISA ☐ MASTERCHARGE ☐
_____ Exp. Date _____

Name _____

Address _____ Signature _____

City, State, Zip _____ ☐ Send me your catalog, Brainfood.

☐ **Yes, send me your free catalog, BRAIN FOOD, which lists over 150 microcomputer books covering software, hardware, business applications, general computer literacy and programming languages.**

NAME _____

ADDRESS _____

CITY, STATE, ZIP _____

☐ **Yes, send me your free catalog, BRAIN FOOD, which lists over 150 microcomputer books covering software, hardware, business applications, general computer literacy and programming languages.**

NAME _____

ADDRESS _____

CITY, STATE, ZIP _____

☐ **Yes, send me your free catalog, BRAIN FOOD, which lists over 150 microcomputer books covering software, hardware, business applications, general computer literacy and programming languages.**

NAME _____

ADDRESS _____

CITY, STATE, ZIP _____

FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG

Mail To:

dilithium Press
P.O. Box E
Beaverton, OR 97075

Or Call:

800-547-1842

FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG FREE CATALOG FREE CATALOG FREE CATA LOG FREE CATALOG

This workbook is an excellent tool for teaching children how to program a microcomputer in BASIC. It's an individualized, self-paced approach that accompanies the book **AN ATARI FOR KIDS**.

Most activities in this workbook can be done without a computer, so can be used as seatwork (solving the problem of 1 computer and 25 students).

Written approximately at the 4th grade reading level, this workbook:

- Includes worksheets to go along with each chapter of **AN ATARI FOR KIDS**
- Approaches programming in a fresh, fun way
- Is easy for kids to understand
- Contains illustrations interesting to kids
- Provides practice and reinforcement for programming skills your students learn
- Has activities that focus on problem solving, improved thinking skills and creativity
- Concludes each unit with fun pages that test your students' learning progress

AN ATARI FOR KIDS and
AN ATARI IN THE CLASSROOM:
Teacher's Guide also available.

dilithium Press
PROGRAMMING